The Dissertation Committee for Li-San Wang
certifies that this is the approved version of the following dissertation:

Large-Scale Phylogenetic Analysis

Committee:

Tandy J. Warnow, Supervisor

Inderjit S. Dhillon

Robert K. Jansen

C. Gregory Plaxton

Michael A. Steel

Large-Scale Phylogenetic Analysis

by

Li-San Wang, B.S., M.S.

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

May 2003

To my parents.

Acknowledgments

This work would not be possible without the help and support from many people. First I would like to express my gratitude to my advisor, Professor Tandy Warnow. Tandy is the advisor any Ph.D. student dreams of: easygoing, thoughtful, caring, always an avid researcher with deep insights, and an enthusiastic planner for her students' careers. I have learned a great deal from her, both as a scientist and a person.

I benefited greatly from Professors Bob Jansen and Bernard Moret, for their advices and instructions on my research. They are always nice and patient when I describe my ideas to them, and their guidance helped me better appreciate the various aspects of computational biology – it is almost like I have two more advisers, for which I am very grateful.

I collaborated with Professor Nina Amenta for almost a year on the Tree Visualization Project. The project is so groundbreaking in phylogenetics, the research opportunity is almost unlimited; we had a great time brainstorming the various research directions.

Comments from Professors Mike Steel, Inderjit Dhillon, and Greg Plaxton were very helpful in making this dissertation better. Though Mike did not make it to my defense, he thoroughly read my dissertation and gave me detailed comments. I met him twice at the States; he gave me many invaluable suggestions regarding my research. Before Tandy became my advisor, I studied clustering algorithms with Greg for two semesters. Though I did not produce significant results, this experience was very educational to me.

I owe a lot to the friendly fellow students from Tandy's lab: Usman Roshan, Luay Nakhleh, Cara Stockham, Ganeshkumar Ganapathysaravanabavan, and Jerry Sun. They have been always helping and nice, and the discussions with them were always interesting and useful. The Department of Computer Sciences at the University of Texas is a wonderful environment to work at. The two administrators, Laurie Alvarez and Gem Naiver, helped me tremendously with many administrative affairs even though I am only a student. Jerry and Laurie also helped my dissertation a great deal: Laurie spent her own time proofreading this dissertation and caught many mistakes, and Jerry helped me turning in my dissertation for my defense while I was in New Jersey.

I am indebted to my wife, Shin-Yi Chou, for her love and unwavering support. For three years, she worked in New Jersey while I pursued my Ph.D. degree at Texas. I admire her firm courage, without which this kind of life style would not be possible. Our daughter, Erica, was born on December 9 2002, soon after I defended my dissertation. May she find life full of surprises and jubilation like I do. Thanks to Shin-Yi's parents, who cared for Shin-Yi and Erica while she and I were 1,500 miles apart most of the time. Finally, I dedicate this work to my parents, Mr. Tse-Chai Wang and Ms. Chin-Fang Li. They devote most of their life to further my education and better my personality; they are always my inspiration to better myself.

Large-Scale Phylogenetic Analysis

Publication No. _____

Li-San Wang, Ph.D.
The University of Texas at Austin, 2003

Supervisor: Tandy J. Warnow

The phylogeny problem is to reconstruct the phylogenetic tree in which the leaves are labeled by the taxa we are interested in, and the internal nodes are ancestral taxa. Recent advances in molecular biology and genomics have provided biologists with molecular data at an unprecedented rate and scale; in particular whole genome data for more and more species. First, the number of possible phylogenetic trees grows superexponentially with the increase of the number of species being studied. Second, detailed sequence data for each species usually convey conflict. Third, more species usually means more evolutionary events along the evolutionary tree. This usually leads to highly saturated data, which are difficult to analyze in general.

In this thesis I present two possible approaches to solve this difficulty. The first approach is to use genome rearrangement evolution, an evolutionary process that has lower evolutionary rate than DNA sequence evolution. The second approach is to process multiple trees returned by tree reconstruction algorithms by applying clustering methods.

Table of Contents

Chapter 1

Introduction

1.1 Difficulties in Large-Scale Phylogeny Reconstruction

The phylogeny problem is to reconstruct the *phylogenetic tree* in which the leaves are labeled by the taxa (species, genes, *etc.*) we are interested in, and the internal nodes are ancestral taxa. The phylogenetic tree reflects the evolutionary history about the leaf species: each internal node is a speciation event in the evolutionary history. We assume no hybridization occurred, i.e. at any point two species do not interbreed to create a new species. Therefore the evolutionary history can be represented as a tree. Phylogenetic trees are important since they reveal the relationships between different species and genes that provide insights and promote advance in molecular genetics, medicine, and drug design.

Recent advances in molecular biology and genomics have provided biologists with molecular data at an unprecedented rate and scale; in particular whole genome data for more and more species. For the first time we are now in a position to be able to study the evolutionary history of thousands of species at the same time. New approaches are necessary because of the following reasons. First, the number of possible phylogenetic trees grows superexponentially with the increase of the number of species being studied. Second, detailed sequence data for each species usually convey conflict; for example,

different genes may indicate different evolutionary histories that contradict each other. As a result biologists need effective means to process multiple trees produced by a single phylogenetic analysis.

In this thesis I present two approaches to the problem of analyzing data close to saturation. The first approach is to use a different source of data, in particular the change in gene orders, that has lower evolutionary rates than DNA sequence evolution in the same duration. The second approach is to analyze the usually enormous set of candidate trees returned by phylogeny reconstruction using clustering algorithms. The necessary background information for the study of phylogeny reconstruction is provided in Chapter 2.

1.2 First Approach: Genome Rearrangement Phylogeny

The research community has sought other sources of phylogenetic signal, characters that evolve slowly or have a large number of states, since such characters generally have a higher signal-to-noise ratio than the usual 4-states-per-character (site) aligned DNA sequences. One source of characters for phylogenetic analysis is the category of "rare genomic changes" [64]. Rare genomic changes are defined as large-scale mutational events in genomes; among many possibilities are *genomic rearrangements*, which include both gene duplications [46] and changes in gene order [62]. The relative rarity of genomic rearrangements makes these attractive as phylogenetic data. Although it has been suggested that there are not enough genomic rearrangements to provide sufficient numbers of characters for resolving phylogenetic relationships in most groups (e.g. chloroplast genomes [58]), increased genome sequencing efforts are uncovering many new genome rearrangements for use in phylogeny reconstruction.

1.2.1 Distance-based reconstruction with equal gene content

The genomes of some organisms have a single chromosome or contain single chromosome organelles (such as mitochondria [9, 59] or chloroplasts [58, 62]) whose evolution is largely independent of the evolution of the nuclear genome for these organisms. Many single-chromosome organisms and organelles have circular chromosomes. Gene maps and whole genome sequencing projects can provide us with information about the ordering and strandedness of the genes, so the chromosome can be represented by an ordering (linear or circular) of signed genes (where the sign of the gene indicates which strand it is located on). The evolutionary process on the chromosome can thus be seen as a transformation of signed orderings of genes. The process includes events that preserve the gene content of a genome such as inversions, transpositions, and inverted transpositions, and events that change the gene content of a genome such as insertions, duplications, and deletions.

Let T be the true tree on which a set of genomes has evolved. Every edge e in T is associated with a number k_e, the actual number of rearrangements along edge e. The *true evolutionary distance* (*t.e.d.*) between two leaves G_i and G_j in T is $k_{ij} = \sum_{e \in P_{ij}} k_e$, where P_{ij} is the simple path on T between G_i and G_j. If we can estimate all k_{ij} sufficiently accurately, we can reconstruct the tree T using very simple methods, and in particular, using the neighbor joining method (NJ) [4, 65]. Therefore, estimates of pairwise distances that are close to the true evolutionary distances will in general be more useful for evolutionary tree reconstruction than edit distances (the minimum number of changes required to transform one genome to the other), because edit distances *underestimate* true evolutionary distances, and this underestimation can be very significant as the number of rearrangements increases [34, 79] (see

Figure 3.1).

There are two criteria for evaluating a *t.e.d.* estimator: how close the estimated distances are to the true evolutionary distance between two genomes, and how accurate the inferred trees are when a distance-based method (e.g. neighbor joining) is used in conjunction with these distances. The importance of obtaining good *t.e.d.* estimates when analyzing DNA sequences (under stochastic models of DNA sequence evolution) is understood, and well-studied [79]. However, very little work has been done on obtaining *t.e.d.* estimates between whole genomes; only the special case of estimating the actual number of inversions between genomes from the breakpoint distance was solved before this thesis [18].

In Chapter 3, I introduce several *t.e.d.* estimators for genome rearrangement, including `Exact-IEBP`, `Approx-IEBP`, and `EDE`. I then present the simulation results. The new *t.e.d.* estimators, when used with neighbor joining, yield more accurate trees than breakpoint and inversion distances.

1.2.2 Estimating the variances of genomic distances

In Chapter 3, the expected breakpoint distance between G and G' when G' is the genome obtained from G by applying k rearrangements according to the GNT model is obtained as a sum of $O(n)$ terms that we do not know yet how to further simplify. As for the inversion distance, even the expectation is still an open problem.

Estimating the variance of breakpoint and inversion distances is important for several reasons. Based on these estimates we can compute the variances of the `Approx-IEBP` and `Exact-IEBP` estimators (based on the breakpoint distance), and the `EDE` estimator (based on the inversion distance). It is

also informative when we compare estimators based on breakpoint distances to other estimators, e.g. the inversion distance and the `EDE` distance. Finally, variance estimation can be used in distance-based methods to improve the topological accuracy of tree reconstruction.

In Chapter 4 I study the variance of genomic distances. The derivation of the variances of the breakpoint and `IEBP` distances is analytical, but the variances of the inversion and `EDE` distances are obtained by applying numerical methods to simulated data. The result is used with two variants of neighbor joining, BioNJ and Weighbor, that require the variances of the input pairwise distances.

In Chapter 5 I present the results of simulation study of the performance (in terms of the topological accuracy of the output trees) of different distance-based phylogenetic methods, as well as of the quality of the different estimates of the true evolutionary distances and their variances. Of all the methods studied, the combination of `Weighbor` and `EDE` (and its variance) yields the most accurate trees. I also studied the robustness of two methods, `NJ(Exact-IEBP)` and `Weighbor-IEBP`, because they require the parameters of the model as input, which are often unknown.

1.2.3 Parsimony-based phylogeny with equal gene content

Let us be given a distance measure D between any two genomes. Let G and G' be the two genomes at the endpoints of an edge e, then the length of e is $D(G, G')$. The *length* of a tree T where all nodes are labeled by genomes is defined as the sum of the lengths of all edges of T. For genomes with equal gene content (i.e. gene orders) we can use the breakpoint distance, inversion distance, transposition distance, or even the `EDE` and `IEBP` distances. The

parsimony score of T with respect to D is the minimum length over all possible labelings of the internal nodes. The *Maximum Parsimony on Rearranged Genomes* problem asks for the tree topology T that has minimum parsimony score with respect to D. The inversion phylogeny is an example that uses the inversion distance in computing the tree length. Some of the most natural problems include inversion and transposition phylogenies, depending on the evolutionary model the biologists use (inversion-only, transposition/inverted transposition only, and inversion/transposition/inverted transposition equally likely); more generally we have the ITT phylogeny where the distance is the weighted edit distance using inversions, transpositions and inverted transpositions. However, we do not know how to solve these problems efficiently: the inversion phylogeny is NP-hard [17], and computing the transposition distance alone has unknown computational complexity and is conjected to be NP-hard [6].

The *breakpoint phylogeny* problem was first proposed by Sankoff *et al.* [66], where a tree with minimal breakpoint length is a breakpoint phylogeny. When the breakpoint distance is linearly correlated with the actual number of events, minimizing the number of breakpoints also minimizes the total number of evolutionary events; Blanchette *et al.* [9] observed such a relationship in a group of metazoan mitochondrial genomes. The breakpoint phylogeny is less model-dependent: all it requires is that a single event only creates (or removes) a small constant number of breakpoints, so the correlation between the breakpoint distance and the actual number of events is somehow maintained.

The other advantage of the breakpoint phylogeny is its lower computational cost. Computing the breakpoint phylogeny is also NP-hard for just three genomes [60] even when the number of taxa is 3 (the problem is called *Median*

Problem for Breakpoints (MPB)). However, Blanchette *et al.* reduced MPB to the traveling salesman problem (a well-studied NP-complete problem with excellent and fast heuristics) and developed the software suite `BPAnalysis` to approximate the breakpoint phylogeny; this approach was subsequently refined and enormously accelerated by Moret *et al.* with the `GRAPPA` software suite [51]. These approaches a have running time that is exponential in both the number of genes, and in the number of genomes, because (1) estimating the length of each tree takes time exponential in the number of genes, and (2) these approaches explicitly estimate the length of *every* possible tree topology, and the number of trees is exponential in the number of genomes.

There are several methods that address this issue, which I study in Chapter 6. These methods are still computationally expensive, by comparison to distance-based methods, but are much faster than `GRAPPA` and `BPAnalysis`, because they are exponential in only the number of genomes and not also in the number of genes. They operate by encoding the original input in such a way that standard maximum parsimony heuristics (for aligned sequences) can be applied to the encoded data, thus greatly speeding up the search (because calculating a tree length under this encoding of the data is then polynomial in the number of genes, rather than exponential). Another appealing property of these methods is that they can be seen as heuristics for the breakpoint phylogeny, since under certain conditions the optimal breakpoint phylogeny for the original data will be an optimal solution to the maximum parsimony problem on the encoded data. I examine the performance of these methods in Chapter 6.

At the end of Chapter 6, I present the results of a simulation study examining the relationship between topological accuracy and two definitions of

tree length: the number of breakpoints on the tree and the number of inversions on the tree. We find that both definitions for tree length are correlated with topological accuracy, with the correlation weakest for genomes of 37 genes (the mitochondrial genome), especially when the dataset is close to saturation.

1.3 Second Approach: Postprocessing of Phylogeny Reconstruction

The second approach is to extract information from the set of trees returned by tree reconstruction algorithms such as maximum parsimony. The old approach is to return a consensus tree as a representative of the set of inferred (most parsimonious) trees. The approach has the following two drawbacks. First, some consensus methods are sensitive to outlier trees (i.e. a very small number of trees that are remotely similar to all the other trees), which are not uncommon in parsimony searches. For example, the strict consensus (the most popular consensus method) can become very unresolved due to a single tree that is distant from other trees that are very similar to each other. The other problem is the consensus tree does not carry the information about how the set of trees is distributed in the tree space.

1.3.1 Postprocessing using clustering algorithms

Phylogenetic analysis can be divided into three stages. In the first stage, a researcher collects data (such as DNA sequences) for each of the different taxa under study. In the second phase, she applies a tree reconstruction method to the data. Many tree reconstruction methods produce more than one candidate tree for the input dataset. For example, the *maximum parsimony* [79] method returns those binary trees with the lowest parsimony score. (The parsimony

score of a tree is the minimum tree length, i.e. the sum of distances between two endpoints across all edges, obtained by any way of labeling the internal nodes.) Very often the number of trees can be in the hundreds or thousands. In the last phase, the consensus tree of the candidate trees is computed so as to resolve the conflict, summarize the information, and reduce the overwhelming number of possible solutions to the evolutionary history.

Many consensus tree methods are available; we are particularly interested in the strict consensus because of the ease of interpretation. There are several shortcomings of this approach, including loss of information and being sensitive to the input.

In Chapter 7 I present a different approach to postprocessing. The set of candidate trees is divided into several subsets using clustering methods. Each cluster is then represented by its own consensus tree. I pose several theoretical optimization problems for these kinds of outputs, and present some initial progress on these problems; these are presented in Section 7.3. The rest of the chapter is focused on an empirical study, which is presented in Section 7.4.

Chapter 2

Background

2.1 The Phylogeny Problem

2.1.1 Phylogenetic trees and bipartitions of leaves

Phylogenetic trees A *phylogenetic tree* for a set S of taxa is a tree whose leaves are labeled bijectively by S and does not contain any node of degree two except the root if the tree is rooted. In a rooted phylogenetic tree there is a *significant node* called the root, and every edge in the tree is directed away from the root. A phylogenetic tree reflects the evolutionary history of a set of taxa at the leaves in the following way: each of the leaves of the tree is labeled by a distinct known taxon, and each internal node corresponds to a *speciation* event, i.e. new taxa are created along the edges pointing out from the internal node. Given any tree T we can convert it to a phylogenetic tree by removing degree-2 nodes, i.e. nodes to which exactly two edges are incident. The tree(s) inferred by most phylogenetic methods are unrooted phylogenetic tree(s). Given a tree T, we let $V(T)$, $L(T)$ and $E(T)$ be the sets of nodes, leaves, and edges in T, respectively.

Edges and bipartition of leaves By deleting any edge e in T, we create two new subtrees T_1 and T_2. We say the edge e induces a bipartition $L(T_1)|L(T_2)$ of $L(T)$. Note that the order of the two sets of leaves in the bipartition is irrelevant: $L(T_1)|L(T_2)$ and $L(T_2)|L(T_1)$ are equal. The bipartitions corre-

sponding to the external edges (i.e. those edges incident to a leaf) are present in every phylogenetic tree on the same set of leaves, so they are trivial in the sense of providing no information with respect to the tree topology. We call the external edges *trivial* edges and internal bipartitions *nontrivial* edges. A phylogenetic tree with n leaves will have at most $n-3$ internal edges. From now on we will not differentiate between an edge and its induced bipartition unless necessary. When comparing edges of two trees, we say two edges, one from each of the two trees, are equal if their induced bipartitions are equal.

Resolving a phylogenetic tree The *degree of resolution* of a phylogenetic tree is the number of internal edges. A *star* is a phylogenetic tree without any internal edge. A binary phylogenetic tree is *fully-resolved*, since we cannot insert new edges so that the result is still a phylogenetic tree. A tree T_1 *refines* another tree T_2 if $E(T_2) \subseteq E(T_1)$; we will use the notation $T_2 \leq T_1$. The relation of tree refinement is a partial ordering on the set of all phylogenetic trees.

Figure 2.1 contains two examples for the concepts described above.

2.2 Comparing Phylogenetic Trees

2.2.1 The Robinson-Foulds distance and the false positive and negative rates

We now define the *Robinson-Foulds distance* [63] between two unrooted phylogenetic trees. Given a model tree T_1 and an inferred tree T_2 on the same set of leaves S, $|S| = n$, an edge e in T_1 is a *false negative* if it is not in T_2; an edge e in T_2 is a *false positive* if it is not in T_1. The *false positive* and *false negative rates* of T_1 with respect to T_2 are the numbers

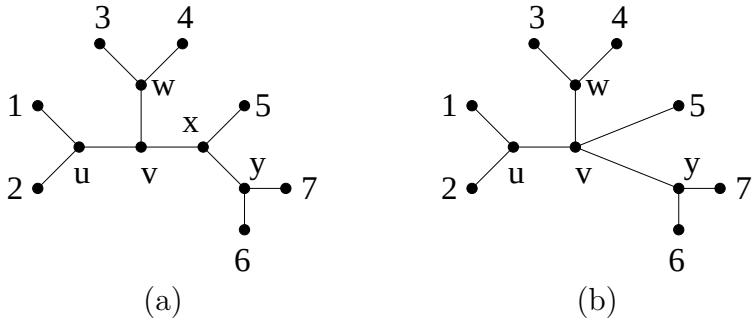


Figure 2.1: Examples of trees. The nodes are highlighted by solid black circles. The leaves are labeled by numerals, and the internal nodes are labeled by letters. (a) An unrooted binary phylogenetic tree. Every internal node has degree 3. The edge (u, v) is an internal edge, and the edge $(1, u)$ is an external edge. The induced bipartition for (u, v) is $\{1, 2\}|\{3, 4, 5, 6, 7\}$. (b) A phylogenetic tree that is unresolved since the degree of node v is 4. Note this tree is refined by the tree in (a).

of false positive and false negative edges, respectively. We denote them by $FP(T_1; T_2)$ and $FN(T_1; T_2)$. We immediately have $FP(T_1; T_2) = FN(T_2; T_1)$. The Robinson-Foulds distance is the sum of false positive and negative rates: $d(T_1, T_2) = FP(T_1; T_2) + FN(T_1; T_2)$. Note that the roles of the model tree and the inferred tree are defined *a priori* by the scientist, but the Robinson-Foulds distance is symmetric: the distance stays the same after interchanging the roles of the two trees. It is straightforward to prove that $d(\cdot, \cdot)$ as defined above as a metric on the set of all trees on the same set of leaves. Finally, if both T_1 and T_2 are binary, $FP(T_1; T_2) = FN(T_1; T_2)$.

2.2.2 Consensus trees

Some phylogenetic analyses, such as maximum parsimony (see Section 2.4.2), return multiple candidates trees with different topologies. There-

fore it is necessary to apply consensus methods that produce a single tree from the large set of candidates under these circumstances. The most known consensus methods are the *strict* and *majority* consensus trees. Let $\mathcal{T} = \{T_1, \ldots, T_m\}$ be the set of distinct trees. If we take the intersection $\cap_{i=1}^{m} E(T_i)$, there exists a tree whose set of edges coincide with this set (just pick a tree from $\mathcal{T}$ and contract edges that are not in this intersection). This tree is called the strict consensus tree of $\mathcal{T}$. In other words, each edge in the strict consensus tree is in every tree in the set of input trees.

For any edge e, let $r_e = |\{T_i | e \in E(T_i)\}|/m$; in other words, r_e is the percentage of input trees that contain edge e. Let $E = \{e | r_e > \frac{1}{2}\}$. It can be shown there exists a unique tree whose edge set is E; this tree is called the majority consensus tree of $\mathcal{T}$. In [49] it is shown the majority consensus $Maj(\mathcal{T})$ minimizes $f(T) = \sum_{T' \in \mathcal{T}} d(T, T')$, the sum of Robinson-Foulds distances to all trees in $\mathcal{T}$, and hence is also a *median* of the set of trees $\mathcal{T}$.

There are other types of consensus trees; the Adams consensus [1] and the Nelson consensus are two examples. In this thesis we only use the strict and majority consensus trees.

2.2.3 Phylogenetic islands

TBR distance In addition to the Robinson-Foulds distance mentioned in Section 2.1, we now describe another topological distance called *TBR distance*.

A tree topological operation (or a move) transforms one topology to another. The TBR distance is defined in terms of a class of tree topological operations called *Tree Bisection and Reconnection (TBR)* [2] moves. A TBR move does the following. An edge (u, v) is removed from tree T to create two

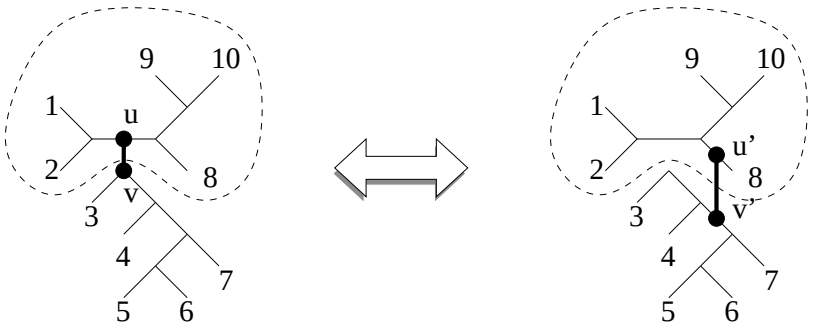


Figure 2.2: The *Tree Bisection and Reconnection (TBR)* move. A TBR move removes an edge (u, v) from the tree to yield two subtrees (separated by the dashed circle); the new tree is obtained by forming a new edge (u', v') to connect the two subtrees.

unrooted subtrees T_u and T_v (u and v may need to be removed to make the two trees valid phylogenetic trees). We then connect the two subtrees to form a new tree T' by inserting a new edge $e' = (u', v')$, where u' is on an edge in T_u and v' is on an edge in T_v (in the case any of the two subtrees are single-node trees, the endpoint is the node itself). Figure 2.2 illustrates the TBR operation.

Given two phylogenetic trees T and T', the *TBR distance* between T and T' is the minimum TBR moves required to apply successively to transform T to T'.

Phylogenetic islands The only method currently used by biologists for partitioning phylogenetic trees into clusters is *phylogenetic islands* [44]. A *phylogenetic island* [44] of a set of trees $\mathcal{T}$ is defined as follows. We create a graph $G(\mathcal{T})$ where each vertex corresponds to a tree in $\mathcal{T}$, and there is an

edge (i, j) between two trees i and j if the two trees are one TBR-move apart (see Section 2.2.3). Each connected component of $G(\mathcal{T})$ is a phylogenetic island (cluster). The significance of this particular clustering method lies in its relation with maximum parsimony and heuristic search. In PAUP* 4.0 [78] the heuristic search implements hill-climbing over the tree space by using TBR moves to modify a given tree topology to obtain new candidate trees. Testing if the TBR distance between any two binary trees on n leaves is one can be done in $O(n)$ time [72].

2.3 Basic Statistical Concepts

Statistics and estimators Let X be a random variable having domain $\mathcal{X}$ and distribution f_X. We say the value of X is an *observation* drawn from the *population* (i.e. the domain of X). A *sample* $\mathbf{X}_n = \{X_1, X_2, \ldots, X_n\}$ is a set of n outcome(s) from X; here the *sample size* is n. The *sample space* of X where the sample size is n is $\mathcal{X}^n$, the set of all possible outcomes of the sample $\mathbf{X}_n$. A *statistic* $T : \mathcal{X}^n \to \mathcal{R}$ on a sample $\mathbf{X}_n$ from X is a real-valued function on the sample space $\mathcal{X}^n$. Note the statistic is not a single function but a family of functions indexed by n, the sample size. An *estimator* $\theta(\mathbf{X}_n)$ for a parameter θ in the distribution of X is a function of $\mathbf{X}_n$. An estimator θ is *based on a statistic* T if θ is a function of T. We usually use $\hat{\theta}$ to denote an estimator for a parameter θ. One goal of the statistics research is to develop estimators for the distribution of the population or its parameters (that are not directly observable) based on the observations (sample), and study the properties of the estimators such as their expectations and variances. In the context of phylogeny reconstruction, we want to estimate the topology (and sometimes the number of events along each edge) of the phylogeny, and the

observations are the characteristics of the taxa at the leaves.

Quality of an estimator An estimator $\hat{\theta}$ for the parameter θ is *unbiased* if $E(\hat{\theta}) = \theta$. The *mean squared error* of $\hat{\theta}$ is $MSE(\hat{\theta}) = E(\hat{\theta} - \theta)^2$. One can show that $MSE(\hat{\theta}) = \text{Bias}(\hat{\theta})^2 + Var(\hat{\theta})$, where $\text{Bias}(\hat{\theta}) = E(\hat{\theta}) - \theta$ is the *bias* of the estimator. When $\hat{\theta}$ is unbiased, $MSE(\hat{\theta}) = Var(\hat{\theta})$.

Asymptotic behavior of an estimator A real-valued estimator $\hat{\theta}$ for parameter θ is *asymptotically unbiased* if $\lim_{n\to\infty} E(\hat{\theta}) = \theta$; $\hat{\theta}$ is *statistically consistent* if for all $\epsilon > 0$, $\lim_{n\to\infty} \Pr(|\hat{\theta} - \theta| < \epsilon) = 1$. One can show that $\hat{\theta}$ is statistically consistent if and only if it is asymptotically unbiased and $\lim_{n\to\infty} Var(\hat{\theta}) = 0$ (see any standard statistics textbook for details). Thus, among a set of estimators to the same parameter, we favor those that are unbiased, and whose variances diminish as the sample size increases. As a side note, when the number of possible states for θ (and hence $\hat{\theta}$) is finite, an equivalent definition for statistical consistency is $\lim_{n\to\infty} \Pr(\hat{\theta} = \theta) = 1$.

Statistical consistency of phylogeny reconstruction methods A phylogeny reconstruction method is an estimator for the tree topology and other aspects of the model, such as the number of events on the edges and internal node states. Let k be the maximum size of the input data for each taxon (e.g., the length of the (aligned) DNA sequences from each taxon). Therefore the observations at the leaves together can be regarded as a random variable, the distribution of which is determined by the true phylogenetic tree and the evolutionary model. If we only consider reconstructing the topology, the method is statistically consistent if the probability the method returns the correct tree

topology approaches 1 as k approaches infinity. In real life, the data size is not infinite.

2.4 Tree Inference

The three main approaches for tree inference are (1) distance-based methods, (2) maximum parsimony, and (3) maximum likelihood. In this thesis we focus on distance-based methods and maximum parsimony. For details of the other two approaches see [79].

2.4.1 Distance-based methods

Additive matrices and distance correction Given the set of taxa and their observed characteristics, we can define distances between any pair of taxa and calculate the distance matrix of the set of taxa. However, such distances may underestimate the actual number of events between any two taxa along the true phylogeny. For example, let us look at the number of different sites between the DNA sequences of any two taxa. Since the changes occur randomly and can sometimes cancel one another, the actual number of changes must be greater than this "minimum" distance. See the following example of 3 DNA sequences:

$$\begin{array}{ccc} \mathrm{X} & = & \texttt{ATTACTG} \\ \mathrm{Y} & = & \texttt{CTTATAC} \\ \mathrm{Z} & = & \texttt{ATGCCAA} \end{array}$$

Assume somehow we know X evolves into Y, and Y evolves into Z. The number of sites changed between X and Y is 4, and the number of sites changed between Y and Z is 5. However the number of sites changed between X and Z is 4, which is lower than the total number of changes, 9. A closer

look reveals that certain sites have more than one change, but their states in X and Z are either different or identical. In the second case the change is lost if we do not have the knowledge about Y, but if the sites are different all we can tell is that at least one change occurred but not the exact number of changes.

If we can estimate the number of actual changes based on the knowledge of the evolutionary process, we usually can improve the topological accuracy of the inferred tree. This idea is well supported by simulation and real data studies for DNA sequence data [79].

Neighbor joining Neighbor joining [65] is the most popular distance-based tree inference method. The input to the method is an $n \times n$ distance matrix D on n leaves. Initially each leaf is in its own subtree. The algorithm iteratively chooses a pair of subtrees that is most likely to be siblings according to the distances between all roots of the subtrees; the criterion is listed in line 2(a)–(c) in the algorithm in the next paragraph. A new subtree is created by making these trees subtrees of the new tree's root, and the distances between the new root and roots of other subtrees are updated. The loop stops when there are only two subtrees left; the algorithm then returns the rooted tree T by joining the roots of the two subtrees to the root of T.

The neighbor joining algorithm is listed as follows:

1. **Initialization:** Set tree T_i to be the rooted tree having the i^{th} taxa as its only node. Set $\mathcal{F} \leftarrow \{T_1, T_2, \ldots, T_n\}$. Set $k = n + 1$.
2. **Do**
   (a) Define $S = \{x | \exists T \in F \text{ s.t. } x \text{ is the root of } T\}$.

(b) For each node r in S, compute $u_r = \sum_{j \in S, j \neq r} \frac{D_{ij}}{|S|-2}$.

(c) Choose $i, j \in S$ such that $D_{ij} - u_i - u_j$ is minimized.

(d) Create a new tree T_k with root k such that the two child subtrees of k in T are the two trees with root i and j. The length of the two new edges are:

$$l_{ik} = \frac{1}{2}D_{ij} + \frac{1}{2}(u_i - u_j), \quad \text{and} \quad l_{jk} = \frac{1}{2}D_{ij} + \frac{1}{2}(u_j - u_i)$$

(e) Compute the distance from k to other nodes in S:

$$D_{kl} = \frac{1}{2}(D_{il} + D_{jl} - D_{ij}) \quad \forall l \in S, l \neq i, j$$

(f) Delete T_i, T_j from $\mathcal{F}$.

(g) $k \leftarrow k + 1$.

until only one tree is left in $\mathcal{F}$.

3. Output the tree.

The path $P(i, j)$ between two leaves i, j on T is the set of edges such that for every $e \in P(i, j)$, i and j are at different sets of the bipartition π_e induced by e. Each $e \in E(T)$ has length α_e, the *actual* number of events on the edge.

Given a tree T with edge lengths, we define the tree distance between the leaves as follows: the distance between any two leaves is the sum of the lengths of the edges on the simple path connecting the two leaves in the tree. The following theorem shows when the input distance matrix D is close to the tree distance between the leaves, neighbor joining returns the correct tree topology:

Theorem 1. *(From [4]) Let T be the true tree with edge length. Assume T is binary, and let α be the tree distance between the leaves defined by T. Let $x = \min_{e \in E(T)} l_e$. If $\min_{i,j \in L(T), i \neq j} |D_{ij} - \alpha_{ij}| < \frac{x}{2}$, then neighbor joining with input D returns a tree T' having the same topology as T (i.e. there are no false positives or false negatives in T' with respect to T).*

Note that neighbor joining always return binary trees. In [4] the statistical consistency of neighbor joining under the Generalized Markov model for DNA sequences is established.

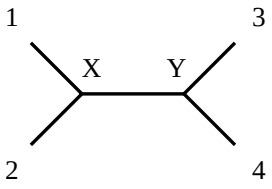
In Chapter 4 we will use two modified versions of neighbor joining called `BioNJ` [25] and `Weighbor` [11]. Both methods use the variance of the tree distance estimators in the distance update steps to improve the accuracy of the tree reconstruction.

2.4.2 Maximum parsimony

The *maximum parsimony* approach tries to find the tree topology that minimizes the *parsimony score*. In this section we use DNA sequence parsimony to illustrate the idea, though the concept can be extended to other types of data such as gene order (see Chapters 3 and 6).

Consider a set of n taxa $\{1, 2, \ldots, n\}$. Each taxon has a DNA sequence of length k. Fix any (unrooted) tree topology T (with leaves labeled by the n taxa). Tree T has $n - 2$ internal nodes. Thus, there are $(n-2)4^k$ ways of assigning sequences of length k to these $n - 2$ internal nodes[1]. For every way of assigning sequences, we can compute the the parsimony score of this

[1]Here 4 is the number of different nucleotides in a DNA sequence: A (adenine), C (cytosine), T (thymine), and G (guanine).




```
1:  AATATCAC    3:  CCCTCATG
2:  AACTATAC    4:  CACACTTG
X:  AACAATAC    Y:  CACACTTG
```

Figure 2.3: Example for parsimony score for DNA sequence data. Left: the most parsimonious tree topology. Right: the DNA sequences of the taxa and an optimal sequence assignment for internal nodes. The parsimony score is 11 when the cost of a mismatch is 1.

particular assignment by summing the Hamming distances (i.e. the number of different sites between two sequences) between two endpoints of every edge in T. The *length*, or the parsimony score of T is the parsimony score of the assignment that has minimal score. See Figure 2.3 for an example.

For the case of DNA sequence parsimony, it can be shown that for any unrooted tree topology T the parsimony score is the same no matter how we root T; the intuition is the Hamming distance between two sequences is symmetric, so the distance between two endpoints of any edge is not changed no matter how we root the tree (and change the parent-child relationship between the two endpoints of the edge). This is why maximum parsimony alone cannot root the phylogeny for DNA sequence evolution.

The parsimony score of any fixed tree topology for DNA sequences can be computed in time linear in the size of input sequences. The following theorem is folklore in the phylogenetic analysis community:

Theorem 2. *[24, 29] Let n be the number of leaves in the unrooted binary tree T, and let k be the sequence length. If each site has at most s different states, then the parsimony score for T can be computed in $O(snk)$ time when the cost*

for any mismatch in each site is 1. Furthermore, if we allow an arbitrary cost function for mismatches, then we can compute the parsimony score in $O(s^2nk)$ *time.*

We can compute the most parsimonious tree by comparing the parsimony scores of all tree topologies. However the number of topologies is superexponential in the number of leaves. Branch-and-bound heuristics are often used to accelerate the brute force approach [78], but there is no guaranteed improvement in the time complexity. Another approach is to use a hill climbing heuristic. In each search run a starting tree is selected. We then perform operations such as TBR that change the tree topology while lowering the parsimony score at the same time, until a minimum is reached. The search can be repeated several times with different starting trees to avoid being trapped in a bad local minimum. By avoiding exhaustive searches, this is currently the only practical approach for datasets with large numbers of taxa.

The maximum parsimony approach is based on the assumption that evolutionary events are rare, so that a phylogeny along with the ancestral states that has more mutations should have a *much* lower probability than a phylogeny that has fewer mutations. Maximum parsimony is not statistically consistent for DNA sequence evolution; the famous example is the Felsenstein zone (see Figure 2.4). Despite its lack of statistical soundness, maximum parsimony is still one of the most popular methods in the biological community due to its simplicity and (relative) lack of assumptions about the underlying evolutionary model [75].

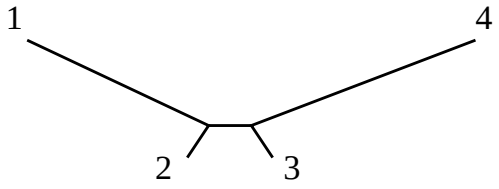


Figure 2.4: The Felsenstein zone. The lengths of the middle edge and the two edges incident to leaves 2 and 3 are small compared to the lengths of the two edges incident to leaves 1 and 4. Under the Jukes-Cantor model (along each edge of the tree on which the evolution takes place, substitution events (changes) in each site of the DNA sequence obey a Poisson process whose mean number of events is (and identical for each site) the length of edge, and the probability of changing from any state to any other state is equal; see [79] for details), both neighbor joining using the corrected distances and maximum likelihood method correctly reconstructs the topology as 12|34 when the sequence length is infinite, while the maximum parsimony returns the topology 14|23 with probability 1.

2.5 Steps of a phylogenetic analysis

A complete phylogenetic analysis can be divided into the following four stages:

1. **Data collecting stage**. Characteristics of the taxa we study are collected for use in the subsequent stages to reconstruct the phylogenetic tree(s) of the species. This stage is the least algorithmic from the perspective of computational phylogenetic analysis. Common types of characteristics include the appearance of the taxa (e.g. presence of wings, weight and size, *etc.*), comparison of biological functions possessed by the species at the leaves (e.g. bones, muscles, warm or cold blood, *etc.*), and so on.

   With the advent of fast and large-scale techniques molecular genetics,

in recent decades biologists more often use molecular data, such as the DNA sequences of genes common in the taxa under study, and the physical orders of genes on the chloroplast chromosomes in plants or the mitochondrial chromosomes.

2. **Preprocessing stage**. In this stage the data are processed so they are legitimate inputs for the next stage, tree inference. For example, DNA sequences from different species usually have different lengths. These sequences must be aligned before maximum parsimony as the algorithm requires sequences of the same length. Distance-based methods such as neighbor joining require the distances between every pair of taxa be computed.

3. **Tree inference stage**. In this stage, biologists run tree reconstruction algorithm(s) on the preprocessed data and obtain phylogenetic tree(s). Some algorithms, such as neighbor joining, return one tree, and some algorithms, such as maximum parsimony, can return multiple trees. In fact, these algorithms can be run on different sets of data from the same set of taxa.

4. **Postprocessing stage**. In the final stage, biologists summarize the results returned by tree inference. When more than one tree topologies are returned, biologists use the consensus trees such as the strict and/or majority trees to resolve the conflicts. For the set of trees returned by a maximum parsimony analysis, biologists sometimes return the phylogenetic island(s) as well.

In Chapters 3, 4, and 6, I focus on the second and third stages when the data are gene orders; in Chapter 7, I study the last stage of phylogenetic

analysis.

2.6 A question of methodology: why simulation studies are important

Simulation is important for the study of phylogenetics, and simulation studies abound in the phylogenetics literature [11, 25, 30, 34, 69, 73]; see also [31, 32] for a discussion on simulation studies in phylogenetics.

There are at least two important advantages for conducting simulation studies in phylogenetics. First, in real life the true phylogenetic tree is never known since we cannot observe the evolutionary process that happened in the past. In a simulation, the true phylogeny is known beforehand, so we can actually compare it with the tree(s) returned by the algorithm. Second, most analytical approaches in the study of phylogeny are very complicated, and their results tend to be loose. For example, Theorem 1 only tells us a sufficient condition for recovering the whole tree, but it does not tell us how the reconstructed tree looks when the condition is not satisfied. Simulations avoid this problem of looseness by giving actual figures as to how accurate the reconstructed topology is, and they can always be performed when analytical results seem difficult to obtain.

In this thesis I use a lot of simulation studies to discover properties of the different phylogeny reconstruction algorithms. I now briefly describe how a simulation study is conducted.

Consider a new maximum parsimony algorithm A for DNA sequence phylogeny reconstruction. We make the following assumptions to facilitate the simulation.

1. Assume the evolution proceeds in the manner described by a stochastic model M; we are given a phylogenetic tree T with edge lengths, where the length of each edge indicates the expected number of events along the edge. The model M can be parameterized, and we can vary these parameters to observe how they affect the performance of the algorithm being studied.

   In this example, we assume the model M is a stochastic process that, given a phylogenetic tree, simulates nucleotide substitutions along the tree and produce DNA sequences at the leaves. The parameter for M is the ratio γ of the chance of having a transition (an A-G or C-T change) and having a transversion (an A-C, A-T, G-C, or G-T change).

2. We also need a way to generate trees as inputs to M; we call this process Q. We either can generate trees randomly (using a birth-death process or generate uniformly random topologies, for example), generate trees of a specific class of topologies (such as a quartet, i.e. the unrooted tree with four taxa, or complete binary trees), or we can use readily available trees, like trees produced by other biologists.

   In this example, we assume Q generates trees as follows. First, a tree topology is generated uniformly randomly, then each edge in the tree is assigned an edge length of l. Another parameter for Q is the number of taxa, n.

The simulation proceeds in the following manner:

1. Pick a set of values for γ, l, and n. Let them be S_γ, S_l, and S_n, respectively.

2. For every (or some) combinations of settings $\gamma \in S_\gamma$, $l \in S_l$, and $n \in S_n$, we generate K repeated *runs*. Each run is divided into several stages similar to those in a normal phylogenetic analysis:

   (a) (Data generation) A phylogenetic tree $T = Q(l, n)$ is generated using Q with values of l and n. The other part of the input is γ for M. The model M then simulates the evolutionary process, and at the leaves of the tree we obtain DNA sequences. We denote the data by $M(\gamma, T)$.

   (b) (Preprocessing) The sequences are already in the format for A in this case; no preprocessing is necessary.

   (c) (Tree inference) The data is then fed to the algorithm A to produce trees $T_1, T_2, \ldots, T_m$.

   (d) (Postprocessing) In the case of multiple trees returned by A, the consensus tree(s) are used in the comparison instead.

   (e) (Tree comparison) We then compare the topology of the tree computed by A with the original tree T and find out how A performs for reconstructing the topology of T. For example, we can use the Robinson-Foulds distance between the two trees.

3. We then average the error over all K runs to find out how accurate topologically the tree(s) reconstructed by A are for the particular settings γ for M, and l and n for Q. The larger K is, the more statistically significant the results are, and the longer the running time is.

Note that $T = Q(l, n)$ and $M(\gamma, T)$ are both random variables in the simulation, and different runs are independent outcomes of these random variables.

Second, the more different settings we use for l, n, and γ are and the larger K is, the more statistically significant the results are in showing how these parameters affect A, but they are usually limited due to the computational resource and running time available. Finally, we see the stages are different in this case. Instead of data collection we generate the data explicitly, and we have a tree comparison stage at the end.

Quantifying error Given an inferred tree, we compare its "topological accuracy" by computing "false negatives" with respect to the "true tree" [26, 43]. During the evolutionary process, some edges of the model tree may have no changes (*i.e.* evolutionary events) on them. Since reconstructing such edges is at best guesswork, we are not interested in these edges. Hence, we define the true tree to be the result of *contracting* those edges in the model tree on which there are no changes.

The error measure we use in the simulation study in Chapters 5 and 6 is the *false negative rate*, which is defined as the percentage of internal edges in T that are false negative edges with respect to T'.

Chapter 3

Distance-based Reconstruction for Genome Rearrangement Phylogeny on Equal Gene Content

3.1 Definitions

Representations of genomes If we assign a number to the same gene in each genome, a linear genome can be represented by a signed permutation of $\{1, \ldots, n\}$ — a permutation followed by giving each number a plus or minus sign — where the sign shows which strand the gene is on. A circular genome can be represented the same way as a linear genome by breaking off the circle between two neighboring genes and choosing the clockwise or counterclockwise direction as the positive direction. For example, the following are representations for the same circular genome: $(1, 2, 3), (2, 3, 1), (-1, -3, -2)$. The *canonical representation* for a circular genome is the representation where gene 1 is at the first position with positive sign. The first representation in the previous example is the canonical representation. An unsigned genome obeys the rules of a signed genome, except the signs before the genes are dropped. This is needed when we do not have strand information.

[1]The content of this chapter also appeared in [50, 83, 86].

Genome rearrangement events We are particularly interested in the following three types of rearrangements: inversions, transpositions, and inverted transpositions. Starting with a genome $G = (g_1, g_2, \ldots, g_n)$ an *inversion* between indices a and b, $1 \leq a < b \leq n+1$, produces the genome with linear ordering

$$(g_1, g_2, \ldots, g_{a-1}, -g_{b-1}, \ldots, -g_a, g_b, \ldots, g_n)$$

If $b < a$, we can still apply an inversion to a circular (but not linear) genome by simply rotating the circular ordering until g_a precedes g_b in the representation, since we consider all rotations of the complete circular ordering of a circular genome as equivalent.

A *transposition* on the (linear or circular) genome G acts on three indices, a, b, c, with $1 \leq a < b \leq n$ and $2 \leq c \leq n+1$, $c \notin [a, b]$, and operates by picking up the interval $g_a, g_{a+1}, \ldots, g_{b-1}$ and inserting it immediately after g_{c-1}. Thus the genome G above (with the additional assumption of $c > b$) is replaced by

$$(g_1, \ldots, g_{a-1}, g_b, g_{b+1}, \ldots, g_{c-1}, g_a, g_{a+1}, \ldots, g_{b-1}, g_c, \ldots, g_n)$$

An *inverted transposition* is the combination of a transposition and an inversion on the transposed subsequence, so that G is replaced by

$$(g_1, \ldots, g_{a-1}, g_b, g_{b+1}, \ldots, g_{c-1}, -g_{b-1}, -g_{b-2}, \ldots, -g_a, g_c, \ldots, g_n)$$

The Generalized Nadeau-Taylor model We introduce the *Generalized Nadeau-Taylor (GNT) model* by generalizing the Nadeau-Taylor model [53] to arbitrary mixtures of inversions, transpositions, and inverted transpositions.

In the GNT model, inversions, transpositions, and inverted transpositions can occur on each edge. Different inversions have equal probability, as do different transpositions and inverted transpositions. Each model tree has two parameters α and β, where α is the probability a rearrangement event is a transposition, and β is an inverted transposition; the probability for a rearrangement to be an inversion is thus $1 - \alpha - \beta$. The number of events on each edge e is Poisson distributed with mean λ_e. This process produces a set of signed gene orders at the leaves of the model tree.

Breakpoint distances Another popular distance between genomes is the *breakpoint distance* [8]. The breakpoint distance between two genomes is the number of breakpoints in one genome with respect to the other. Note that this definition is symmetric in the sense we can swap the two genomes and produce the same number of breakpoints. Let genome $G = (g_1, \ldots, g_n)$, and let G' be a genome obtained by rearranging G. The two genes g_i and g_j are *adjacent* in genome G if g_i is immediately followed by g_j in G, or, equivalently, if $-g_j$ is immediately followed by $-g_i$. A breakpoint in G' with respect to G is defined as an ordered pair of genes (g_i, g_j) such that g_i and g_j are adjacent in G', but are not adjacent in G (neither (g_i, g_j) nor $(-g_j, -g_i)$ appear consecutively in that order in G). The breakpoint distance between two genomes G and G' is the number of breakpoints in G' with respect to G (or vice versa, since the breakpoint distance is symmetric). For example, let $G = (1, 2, 3, 4)$ and let $G' = (1, -3, -2, 4)$; there is a breakpoint between genes 1 and -3 in G' (w.r.t. G) but not between genes -3 and -2 in G' (w.r.t. G). The breakpoint distance between G and G' is 2.

Outline of the algorithms The algorithm we develop tries to estimate the true evolutionary distance, i.e. the *actual* number of rearrangements between each pair of genomes according to the breakpoint distance between them.

We first define the following concepts formally. A rearrangement ρ is a permutation of the genes in the genome, followed by either negating or retaining the sign of each gene. Let $G_0 = (g_1, g_2, \ldots, g_k)$ be the signed genome of k genes at the beginning of the evolutionary process. For linear genomes we add the two sentinel genes $g_0 = 0$ and $g_{k+1} = k + 1$ in the front and the end of G_0 that are never moved. For any $r \geq 1$, let $\rho_1, \rho_2, \ldots \rho_r$ be r random rearrangements and let $G_r = \rho_r \rho_{r-1} \ldots \rho_1 G_0$ (i.e. G_r is the result of applying these r rearrangements to G_0). For any pair of genomes G and G' of the same number of genes, let $BP(G, G')$ denote the breakpoint distance between G and G'.

Given any linear genome $G = (g_0, g'_1, g'_2, \ldots, g'_k, g_{k+1})$, where $g_0 = 0$ and $g_{k+1} = k+1$ are the two sentinel genes, we define the function $B_i(G), 0 \leq i \leq k$ by setting $B_i(G) = 0$ if genes g_i and g_{i+1} are adjacent, and $B_i(G) = 1$ if not; in other words, $B_i(G) = 1$ if and only if G has a breakpoint between g_i and g_{i+1}. When G is circular there are at most n breakpoints $B_i(G), 1 \leq i \leq n$. We denote the breakpoint distance between two genomes G and G' by $BP(G, G')$. Let $P_{i|r} = \Pr(B_i(G_r) = 1)$; then $E[BP(G_0, G_r)] = \sum_{i=0}^{n} P_{i|r}$ for linear genomes and $E[BP(G_0, G_r)] = \sum_{i=1}^{n} P_{i|r}$ for circular genomes.

Let α and β be the two parameters in the GNT model. Assume we have a easily computable function $\mathcal{F}_r(\alpha, \beta)$ that estimates $E[BP(G_0, G_r)]$ with sufficient accuracy. The algorithm takes the following form:

For each pair of genomes G and G':

1. Find the breakpoint distance (inversion distance) b between G and G'.

2. Find the integer r such that $|\mathcal{F}_r(\alpha, \beta) - b|$ is minimized. The number r is the estimate of the actual number of rearrangements between G and G'.

We introduce three algorithms that compute the pairwise true evolutionary distances of a set of n circular or linear signed genomes having k genes. All methods assume the above algorithmic form; the major difference is in the function $\mathcal{F}_k(\alpha, \beta)$. The first estimator, called `Exact-IEBP`, computes $E[BP(G_0, G_r)]$ exactly under the Generalized Nadeau-Taylor model in $O(n^2k + n^3)$ time for signed circular genomes, and $O(n^2k + k^7)$ time for signed linear genomes. The second estimator, called `Approx-IEBP`, uses an approximation to $E[BP(G_0, G_k)]$ in $O(n^2k + \min\{k, n^2\} \log k)$ time; thus the `Approx-IEBP` estimator is faster than `Exact-IEBP`. In addition, `Approx-IEBP` is more general; it can be applied to circular/linear and signed/unsigned genomes easily. The third estimator, called `EDE`, estimates $E[INV(G_0, G_k)]$ by nonlinear regression of simulation data. Though there is no theoretical guarantee, `EDE` has best empirical performance in simulation.

3.2 The `Exact-IEBP` Distance Estimator

In this section we present the `Exact-IEBP` distance estimator and its derivation. IEBP stands for "inverting the expected breakpoint distance".

3.2.1 Derivation of the `Exact-IEBP` method

Signed circular genomes We now assume that all genomes are given in the canonical representation. Let R_I, R_T, R_V be the set of all inversions, transpo-

sitions, and inverted transpositions, respectively. We assume the evolutionary model is the GNT model with parameters α and β. Within each of the three types of rearrangement events, all events have the same probability. Under the GNT model, $P_{i|r}$ has the same distribution for all i, $1 \leq i \leq r$. Therefore $E[BP(G_0, G_r)] = kP_{1|r}$. Let $\mathcal{G}_k^C$ be the set of all signed circular genomes, and let $W_k^C = \{\pm 2, \pm 3, \ldots, \pm k\}$. We define the function $K : \mathcal{G}_k^C \to W_k^C$ as follows: for any genome $G \in \mathcal{G}_k^C$, $K(G) = x$ if g_2 is at position $|x|$ with the same sign of x. For example, in the genome $G = (g_1, g_3, g_5, g_4, -g_2)$ we have $K(G) = -5$. Since the sign and the position of gene g_2 uniquely determine $P_{1|r}$, $\{K(G_r) : r \geq 0\}$ is a homogeneous Markov chain where the state space is W_k^C. We will use these states for indexing elements in the transition matrix and the distribution vectors. For example, if M is the transition matrix for $\{K(G_r) : r \geq 0\}$, then $M_{i,j}$ is the probability of jumping to state i from state j in one step in the Markov chain for all i and j in W_k^C.

For every rearrangement $\rho \in R_I \cup R_T \cup R_V$, we construct the matrix Y_ρ as follows: for every i and j in W_k^C, $(Y_\rho)_{i,j} = 1$ if ρ changes the state of gene g_2 from j to i. We then let $M_I = \frac{1}{|R_I|} \sum_{\rho \in R_I} Y_\rho$, $M_T = \frac{1}{|R_T|} \sum_{\rho \in R_T} Y_\rho$, and $M_V = \frac{1}{|R_V|} \sum_{\rho \in R_V} Y_\rho$. The transition matrix M for $\{K(G_r) : r \geq 0\}$ is therefore $M = (1 - \alpha - \beta)M_I + \alpha M_T + \beta M_V$. Let x_r be the probability vector for $K(G_r)$. We obtain $\mathcal{F}_r^E(\alpha, \beta)$ that yields the exact value of $E[BP(G_0, G_r)]$:

$$\begin{aligned}
(x_0)_2 &= 1 \\
(x_0)_i &= 0, \quad i \in W_k^C, i \neq 2 \\
x_r &= M^r x_0 \\
\mathcal{F}_r^E(\alpha, \beta) &\triangleq E[BP(G_0, G_r)] = kP_{1|r} = k(1 - (x_r)_2)
\end{aligned}$$

The result in [67] is a special case where $\alpha = \beta = 0$.

Signed linear genomes When the genomes are linear, we no longer have the luxury of placing gene g_1 at some fixed position with positive sign; different breakpoints may have different distributions. We need to solve the distribution of each breakpoint individually by considering the positions and the signs of both genes involved at the same time. Let $\mathcal{G}_k^L$ be the set of all signed linear genomes, and let $W_k^L = \{(u,v) : u, v = \pm 1, \ldots, \pm k, |u| \neq |v|\}$. We define the functions $J_i : \mathcal{G}_k^L \to W_k^L$, $i = 1, \ldots, k-1$, as follows: for any genome $G \in \mathcal{G}_k^L$, $J_i(G) = (x, y)$ if g_i is at position $|x|$ having the same sign of x, and g_{i+1} is at position $|y|$ having the same sign of y. Therefore $\{J_i(G_r) : r \geq 0\}, 1 \leq i \leq k-1$ are $k-1$ homogeneous Markov chains where the state space is W_k^L. For example, in the genome $G = (g_1, g_2, g_4, g_5, g_6, -g_3, -g_7, g_8)$ we have $L_3(G) = (-6, 3)$ and $L_7(G) = (-7, 8)$. As before we use the states in W_k^L as indices to the transition matrix and the probability vectors. Let $x_{i,r}$ be the probability vector of $L_i(G_r)$. For every rearrangement $\rho \in R_I, R_T$, and R_V, Y_ρ is defined similarly as before (for circular genomes), except the dimension of the matrix is different. We then let $M_I = \frac{1}{|R_I|}\sum_{\rho \in R_I} Y_\rho$, $M_T = \frac{1}{|R_T|}\sum_{\rho \in R_T} Y_\rho$, and $M_V = \frac{1}{|R_V|}\sum_{\rho \in R_V} Y_\rho$. The transition matrix M has the same form as that for the circular genomes: $M = (1-\alpha-\beta)M_I + \alpha M_T + \beta M_V$. Let e be the vector where $e_{(u,v)} = 1$ if $v = u+1$, and 0 otherwise (that is, $e_s = 1$ if s is the state where the two genes are adjacent so there is no breakpoint between them). Therefore

$$\begin{aligned}
(x_{i,0})_{(i,i+1)} &= 1 \\
(x_{i,0})_{(u,v)} &= 0, \quad (u,v) \in W_k^L, (u,v) \neq (i, i+1) \\
x_{i,r} &= M^r x_{i,0} \\
P_{i|r} &= 1 - e^T x_{i,r} = 1 - e^T M^r x_{i,0}
\end{aligned}$$

Since the two sentinel genes 0 and $k+1$ never change their positions and signs, their states are fixed. This means the distribution of the two breakpoints B_0 and B_k depend on the state of one gene each (g_1 and g_k, respectively); hence we can use the results from circular genomes to estimate $P_{0|r}$ and $P_{k|r}$. Under the GNT model they have the same distribution. Then the expected breakpoint distance after r events is

$$\begin{aligned} E[BP(G_0, G_r)] &= \sum_{i=0}^{k} P_{i|r} = 2P_{0|r} + \sum_{i=1}^{k-1} P_{i|r} = 2P_{0|r} + \sum_{i=1}^{k-1}(1 - e^T M^r x_{i,0}) \\ &= 2P_{0|r} + (k-1) - e^T M^r \sum_{i=1}^{k-1} x_{i,0} \end{aligned}$$

We now define the `Exact-IEBP` estimator $\widehat{r}^E(G, G')$ for the true evolutionary distance between two genomes G and G':

Algorithm `Exact-IEBP`

Input: Two genomes G and G' having the same set of distinct genes, GNT model parameter α and β.

Output: Integer $\widehat{r}^E(G, G')$, an estimate of the true evolutionary distance between G and G'.

1. For all $r = 1, \ldots, q$ (where q is some integer large enough to bring a genome to random) compute $\mathcal{F}_r^E(\alpha, \beta)$, where α and β are the GNT model parameters.

2. To compute $r = \widehat{r}^E(G, G') \quad (0 \leq r \leq q)$, we

   (a) compute the breakpoint distance $b = BP(G, G')$, then

   (b) find the integer r such that $\left|\mathcal{F}_r^E(\alpha, \beta) - b\right|$ is minimized.

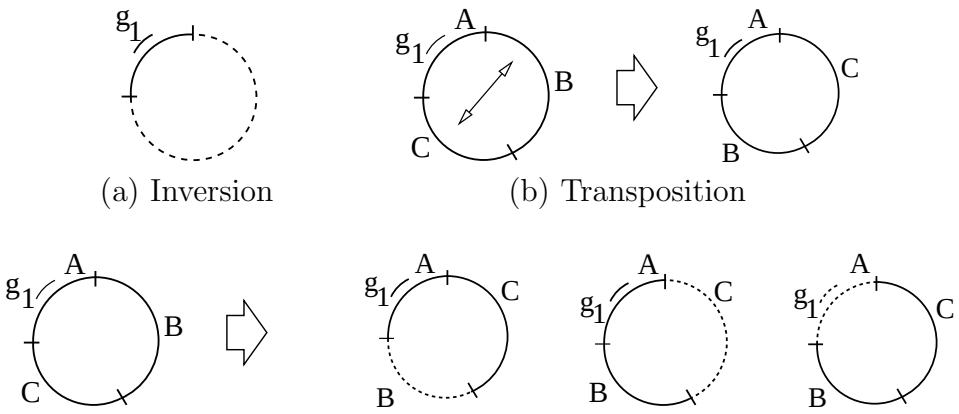
3.2.2 The Transition Matrices for Signed Circular Genomes

We now derive closed-form formulas of the transition matrix M for the GNT model on signed circular genomes with k genes. Let $\binom{a}{b}$ denote the binomial coefficient; in addition, we let $\binom{a}{b} = 0$ if $b > a$. First consider the number of rearrangement events in each class:

1. *Inversions.* By symmetry of the circular genomes and the model, each inversion has a corresponding inversion that inverts the complementary subsequence (the solid vs. the dotted arc in Figure 3.1(a)); thus we only need to consider the $\binom{k}{2}$ inversions that do not invert gene g_1.

2. *Transpositions.* In Figure 3.1(b), given the three indices in a transposition, the genome is divided into three subsequences, and the transposition swaps two subsequences without changing the signs. Let the three subsequences be A, B, and C, where A contains gene g_1. A takes the form (A_1, g_1, A_2), where A_1 and A_2 may be empty. In the canonical representation there are only two possible unsigned permutations: (g_1, A_2, B, C, A_1) and (g_1, A_2, C, B, A_1). This means we only need to consider transpositions that swap the two subsequences not containing g_1.

3. *Inverted Transpositions.* There are $3\binom{k}{3}$ inverted transpositions. In Figure 3.1(c), given the three endpoints in an inverted transposition, exactly one of the three subsequences changes signs. Using the canonical representation, we interchange the two subsequences that do not contain g_1 and invert one of them (the first two genomes right of the arrow in Figure 3.1(c)), or we invert both subsequences without swapping (the rightmost genome in Figure 3.1(c)).



(a) Inversion (b) Transposition

(c) Inverted transposition

Figure 3.1: The three types of rearrangement events in the GNT model on a signed circular genome. (a) We only need to consider inversions that do not invert g_1. (b) A transposition corresponds to swapping two subsequences. (c) The three types of inverted transpositions. Starting from the left genome, the three distinct results are shown here; the broken arc represents the subsequence being transposed and inverted.

For all u and v in W_k^C, let $\iota_k(u,v)$, $\tau_k(u,v)$ and $\nu_k(u,v)$ be the numbers of inversions, transpositions, and inverted transpositions that bring a gene in state u to state v (k is the number of genes in each genome). Then

$$\begin{aligned} M_{u,v} &= (1-\alpha-\beta)(M_I)_{u,v} + \alpha(M_T)_{u,v} + \beta(M_V)_{u,v} \\ &= \frac{1-\alpha-\beta}{\binom{k}{2}}\iota_k(u,v) + \frac{\alpha}{\binom{k}{3}}\tau_k(u,v) + \frac{\beta}{3\binom{k}{3}}\nu_k(u,v) \end{aligned}$$

The following lemma gives formulas for $\iota_k(u,v)$, $\tau_k(u,v)$, and $\nu_k(u,v)$.

Lemma 1. *For all u and v in W_k^C, let $\iota_k(u,v)$, $\tau_k(u,v)$ and $\nu_k(u,v)$ be the numbers of inversions, transpositions, and inverted transpositions that bring a*

gene in state u to state v (k is the number of genes in each genome). Then

$$\iota_k(u,v) = \begin{cases} \min\{|u|-1, |v|-1, k+1-|u|, k+1-|v|\}, & \text{if } uv < 0 \\ 0, & \text{if } u \neq v, uv > 0 \\ \binom{|u|-1}{2} + \binom{k+1-|u|}{2}, & \text{if } u = v \end{cases}$$

$$\tau_k(u,v) = \begin{cases} 0, & \text{if } uv < 0 \\ (\min\{|u|,|v|\}-1)(k+1-\max\{|u|,|v|\}), & \text{if } u \neq v, uv > 0 \\ \binom{|u|-1}{3} + \binom{k+1-|u|}{3}, & \text{if } u = v \end{cases}$$

$$\nu_k(u,v) = \begin{cases} (k-2)\iota_k(u,v), & \text{if } uv < 0 \\ \tau_k(u,v), & \text{if } u \neq v, uv > 0 \\ 3\tau_k(u,v), & \text{if } u = v \end{cases}$$

Proof. The proof of (a) is omitted. This result is first shown in [67].

We now prove (b). Consider the gene with state u. Let v be the new state of that gene after the transposition with indices (a,b,c), $2 \leq a < b < c \leq k+1$. Since transpositions do not change the sign, $\tau_k(u,v) = \tau_k(-u,-v)$, and $\tau_k(u,v) = 0$ if $uv < 0$. Therefore we only need to analyze the case where $u, v > 0$.

We first analyze the case when $u = v$. Assume that either $a \leq u < b$ or $b \leq u < c$. In the first case, from the definition in Section 3.1 we immediately have $v = u + (c-b)$, therefore $v - u = c - b > 0$. In the second case, we have $v = u + (a-b)$, therefore $v - u = a - b < 0$. Both cases contradict the assumption that $u = v$, and the only remaining possibilities that makes $u = v$ are when $2 \leq u = v < a$ or $c \leq u = v \leq n$. This leads to the third line in the $\tau_k(u,v)$ formula. Next, the total number of solutions (a,b,c) for the following two problems is $\tau_k(u,v)$ when $u \neq v$ and $u, v > 0$:

(i) $u < v : b = c - (v-u)$, $2 \leq a \leq u < b < c \leq k+1, u < v \leq c$.

(ii) $u > v : b = a + (u - v),\ \ 2 \leq a < b \leq u < c \leq k + 1, a \leq v < u.$

In the first case $\tau_k(u, v) = (u-1)(k+1-v)$, and in the second case $\tau_k(u, v) = (v-1)(k+1-u)$. The second line in the $\tau_k(u, v)$ formula follows by combining the two results.

For inverted transpositions there are three distinct subclasses of rearrangement events. The result in (c) follows by applying the above method to the three cases. □

3.2.3 Running Time Analysis

Let n be the number of genomes and the dimension of the distance matrix. Since for every pair of genomes we can compute the breakpoint distance between them in linear time, computing the breakpoint distance matrix takes $O(n^2k)$ time. For the purpose of computing all $\binom{k}{2}$ pairwise distances, let m be the number of genomes and the dimension of the distance matrix. We need to compute the distance for at least $O(\min\{n^2, k\})$ distinct breakpoint distance values. Consider the value q, the number of inversions needed to produce a genome that is close to random; we can use this as an upper bound of r in computing the recursion. Both our simulation (see Section 5.1 and Figure 1) and the `Approx-IEBP` formula show that it is reasonable to set $q = uk$ for some constant u that is sufficiently larger than 1 (in our experiment $u = 2.5$ is enough).

Constructing the transition matrix M for circular genomes takes $O(k^2)$ time by Lemma 1. We use the construction in Section 3.2.1 for linear genomes[1].

[1]We believe results similar to Lemma 1 can be obtained for linear genomes, though it is still an open problem.

For each rearrangement ρ, constructing the Y_ρ matrix takes $O(k^4)$ time. Since there are $O(k^2)$ inversions and $O(k^3)$ transpositions and inverted transpositions, constructing the transition matrix M takes $O(k^7)$ time. The running time for computing x_r in `Exact-IEBP` for $r = 1, \ldots, q$ is $O(qk^2) = O(k^3)$ for circular genomes and $O(qk^4) = O(k^5)$ for linear genomes by q matrix-vector multiplications. Since the breakpoint distance is always an integer between 0 and k, we can construct the array $\hat{r}(b)$ that converts the breakpoint distance b to the corresponding `Exact-IEBP` distance in $O(k^2)$ time. Transforming the breakpoint distance matrix into the `Exact-IEBP` distance matrix takes $O(n^2)$ additional array lookups.

We summarize the discussion as follows:

Theorem 3. *Given a set of n genomes on k genes, we can estimate the pairwise true evolutionary distance using `Exact-IEBP` in $O(n^2k + k^3)$ time when the genomes are circular, and $O(n^2k + k^7)$ time when the genomes are linear.*

3.3 The `Approx-IEBP` Distance Estimator

3.3.1 Introduction

The `Exact-IEBP` method estimates the true evolutionary distance by considering a Markov chain with $O(k)$ states, and takes cubic running time in the number of genes per genome. In this section we introduce a variant called `Approx-IEBP` that reduces the running time dramatically by approximating the calculation using lower and upper bounds with provable error bounds.

3.3.2 Extending the model

We use a model more general than the GNT model for the derivation to allow more general results for the `Approx-IEBP` method. We formulate this more general model as follows. A rearrangement class $\mathcal{E}$ acting on $\mathcal{G}_k$ (depending on the context of the problem, $\mathcal{G}_k$ can be either circular or linear) is a pair $(A(\mathcal{E}), f_{\mathcal{E}})$, where $A(\mathcal{E})$ is a set of rearrangements with nonzero probability of taking place, and $f_{\mathcal{E}}(\rho|G)$ is the probability that rearrangement ρ takes place on genome G, for a given $\rho \in A(\mathcal{E})$ and $G \in \mathcal{G}_k$. We say the random variable of rearrangements ρ on genome G is of rearrangement class $\mathcal{E}$ if ρ is in $A(\mathcal{E})$ and has distribution $f_{\mathcal{E}}(\rho|G)$.

Following the notation in Section 3.1, we now present the derivation of `Approx-IEBP`. Assume the rearrangement to act on G is ρ. We make the following definitions:

- $s(i|G, \mathcal{E}) = \Pr(B_i(\rho G) = 1 \mid B_i(G) = 0)$,
- $u(i|G, \mathcal{E}) = \Pr(B_i(\rho G) = 0 \mid B_i(G) = 1)$,
- $\mathrm{Sep}(i|G, \mathcal{E}) = \{\rho \in A(\mathcal{E}) : B_i(\rho G) = 1\}$, and
- $\mathrm{Uni}(i|G, \mathcal{E}) = \{\rho \in A(\mathcal{E}) : B_i(\rho G) = 0\}$.

We focus on rearrangement classes $\mathcal{E}$ where $f_{\mathcal{E}}$ is independent of r and G, and $s(i|G, \mathcal{E})$ is independent of G All three rearrangement classes in the GNT model, namely the class of random inversions, the class of random transpositions, and the class of random inverted transpositions, satisfy these requirements.

We now show the derivation and properties of our *t.e.d.* estimator. We start in Section 3.3.3 with the simple case of rearrangement event classes

where the breakpoints satisfy the Markov property, and find the expected number of breakpoints after k random rearrangements. The result is extended in Section 3.3.4, where the requirement on the Markov property is relaxed; the result is an approximation to the expected number of breakpoints. The error bounds on the approximation are shown in Section 3.3.5. The main result is in Section 3.3.6, where we develop the technique for rearrangement classes that are mixtures of other rearrangement classes. The technique is then applied to the GNT model of genome rearrangements in Section 3.3.7.

3.3.3 Single rearrangement class models where the breakpoints satisfy the Markov property

We start with a simpler case by considering any rearrangement class $\mathcal{E}$ that has the following properties. Assume $s(i|G,\mathcal{E})$ and $u(i|G,\mathcal{E})$ are independent of the past history and the current genome G to be acted upon. Then $\{B_i(G_r)|r \geq 0\}$ is a Markov process (see Figure 3.2), as is shown in the following theorem:

Theorem 4. *Assume $\mathcal{E}$ is a class of rearrangements such that $s(i|G,\mathcal{E})$ and $u(i|G,\mathcal{E})$ do not depend upon genome G. Let their common values be $s(i|\mathcal{E})$ and $u(i|\mathcal{E})$, respectively. Then*

$$P_{i|r} = s(i|\mathcal{E}) \left(\frac{1-(1-s(i|\mathcal{E})-u(i|\mathcal{E}))^r}{1-(1-s(i|\mathcal{E})-u(i|\mathcal{E}))} \right).$$

Proof. We have the following recurrence:

$$\begin{aligned} s(i|\mathcal{E}) &= \Pr(\rho_r \in \mathrm{Sep}(i|G_r,\mathcal{E}) \mid B_i(i|G_r) = 0) \\ &= \Pr(B_i(G_{r+1}) = 1 \mid B_i(G_r) = 0) \end{aligned}$$

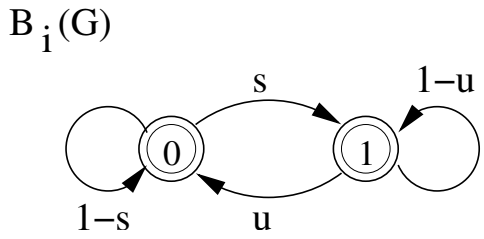


Figure 3.2: Each breakpoint is a two-state stochastic process with two parameters s and u (see Section 3.3.4).

$$
\begin{aligned}
&= \frac{\Pr(B_i(G_{r+1}) = 1 \cap B_i(G_r) = 0)}{1 - P_{i|r}} \\
u(i|\mathcal{E}) &= \Pr(\rho_r \in \mathrm{Uni}(i|G_r, \mathcal{E}) \ \mid \ B_i(G_r) = 1) \\
&= \Pr(B_i(G_{r+1}) = 0 \mid B_i(G_r) = 1) \\
&= \frac{\Pr(B_i(G_{r+1}) = 0 \cap B_i(G_r) = 1)}{P_{i|r}} \\
P_{i|r+1} &= \Pr(B_1(G_{r+1}) = 1) = (1 - P_{i|r})s(i|\mathcal{E}) + P_{i|r}(1 - u(i|\mathcal{E})) \\
&= P_{i|r}(1 - s(i|\mathcal{E}) - u(i|\mathcal{E})) + s(i|\mathcal{E}) \\
P_{i|0} &= 0
\end{aligned}
$$

The proof follows by solving the recurrence. □

Corollary 1. *Let G_r be the result of applying r random inversions to the unsigned linear genome G_0 having k genes. If G_r is linear,*

$$E[BP(G_0, G_r)] = (k-1)\left(1 - \left(\frac{k-3}{k-1}\right)^r\right),$$

and if G_r is circular,

$$E[BP(G_0, G_r)] = \frac{k(k-3)}{k-1}\left(1 - \left(\frac{k-4}{k-2}\right)^r\right).$$

Proof. Follows from Theorem 4, with parameters from Tables 3.1 and 3.2. The linear case is originally in [18] with similar arguments, and the circular case is a simple extension. □

3.3.4 The lower and upper bounds technique for single rearrangement class models

For many other classes of rearrangements, the parameters regarding transitions of $B_i(G)$'s state depend not only on $B_i(G)$ but on other properties of G. For example, the number of inversions that make genes g_1 and g_2 adjacent on signed genomes depend on the number of genes between these two genes. However, for the rearrangement classes $\mathcal{E}$ where $s(i|G, \mathcal{E})$ does not depend on G, we can obtain upper and lower bounds on the expected number of breakpoints and thus *t.e.d.* estimators.

Let $u_{min}(i|\mathcal{E})$ and $u_{max}(i|\mathcal{E})$ be the lower and upper bounds of $u(i|G, \mathcal{E})$ over all genomes G. Observe that a larger value of $u(i|G, \mathcal{E})$ means that genes g_i and g_{i+1} are more likely to be made adjacent, given that they are currently not adjacent. This means $P_{i|r}$, the probability of having a breakpoint between gene g_i and g_{i+1} after r rearrangements, is monotone decreasing on $u(i|G, \mathcal{E})$.

Theorem 5. *Assume $\mathcal{E}$ is a class of rearrangements such that $s(i|\mathcal{E})$ is independent of the genome G currently acted upon. Let $u_{min}(i|\mathcal{E})$ and $u_{max}(i|\mathcal{E})$ be defined as in the previous paragraph. We have $P^L_{i|r} \leq P_{i|r} \leq P^H_{i|r}$ for all r, where*

$$\begin{aligned} P^L_{i|r} &= s(i|\mathcal{E}) \left(\frac{1 - (1 - s(i|\mathcal{E}) - u_{max}(i|\mathcal{E}))^r}{1 - (1 - s(i|\mathcal{E}) - u_{max}(i|\mathcal{E}))} \right) \\ P^H_{i|r} &= s(i|\mathcal{E}) \left(\frac{1 - (1 - s(i|\mathcal{E}) - u_{min}(i|\mathcal{E}))^r}{1 - (1 - s(i|\mathcal{E}) - u_{min}(i|\mathcal{E}))} \right). \end{aligned}$$

Proof. The two recursions determined by $u_{min}(i|\mathcal{E})$ and $u_{max}(i|\mathcal{E})$ can be solved using Theorem 4. The last step is to prove the inequality bounding $P_{i|r}$ by $P^L_{i|r}$ and $P^H_{i|r}$ for all r using induction. When $r = 0$, all three quantities are 0. The induction step is as follows:

$$\begin{aligned} & P^L_{i|r+1} = P^L_{i|r}(1 - s(i|\mathcal{E}) - u_{max}(i|\mathcal{E})) + s(i|\mathcal{E}) \\ \leq \; & P_{i|r}(1 - s(i|\mathcal{E}) - u(i|G_r, \mathcal{E})) + s(i|\mathcal{E}) = P_{i|r+1} \\ \leq \; & P^H_{i|r}(1 - s(i|\mathcal{E}) - u_{min}(i|\mathcal{E})) + s(i|\mathcal{E}) = P^H_{i|r+1} \end{aligned}$$

□

Corollary 2. *Given two random signed circular genomes G and G' on k genes, $k \geq 2$,*

$$E[BP(G, G')] = \frac{k(k - 1.5)}{k - 1}$$

Proof. The expected breakpoint distance between two random genomes is the same as the breakpoint distance between an unrearranged genome G_0 and a random genome G. Under canonical representations, gene 1 is always positive and at the first position in both genomes. Without loss of generality assume gene 2 immediately follows gene 1 in G_0. There are $2(k - 1)$ equally probable choices regarding the sign and position of gene 2 in G, and exactly one of these makes gene 1 and gene 2 adjacent. So the probability of genes 1 and 2 not being adjacent in G is $(2(k - 1) - 1)/(2(k - 1)) = (k - 1.5)/(k - 1)$. The theorem follows since the other $k - 1$ pairs of genes adjacent in G have the same probability of not being adjacent in G as gene 1 and gene 2.

□

This result is apparently new; see [9] for a previous estimate, which this corrects.

Definition 1. *Given any class of rearrangements $\mathcal{E}$ that satisfies the assumption in Theorem 5, we set*

$$\mathcal{F}_r^A(\mathcal{E}) = \sum_{i=0}^{k} \frac{P_{i|r}^L + P_{i|r}^H}{2}.$$

The function $\mathcal{F}_r^A(\mathcal{E})$ is an approximation to the expected number of breakpoints after r random rearrangements drawn from $\mathcal{E}$.

3.3.5 Error bounds of the technique

In this section we bound the absolute and relative errors of the estimator $\mathcal{F}_r^A(\mathcal{E})$ with respect to $E[BP(G_0, G_r)]$. Let

- $R_i^L = 1 - s(i|\mathcal{E}) - u_{max}(i|\mathcal{E})$, and
- $R_i^H = 1 - s(i|\mathcal{E}) - u_{min}(i|\mathcal{E})$.

Note $(R_i^L)^r \leq (R_i^H)^r, \forall r \geq 0$. We now bound the error of the estimator $\mathcal{F}_r^A(\mathcal{E})$.

Lemma 2.

$$\frac{1}{2}(P_{i|r}^H - P_{i|r}^L) \leq \frac{u_{max}(i|\mathcal{E}) - u_{min}(i|\mathcal{E})}{2\ s(i|\mathcal{E})}$$

Proof.

$$\begin{aligned}
& \frac{1}{2}(P_{i|r}^H - P_{i|r}^L) = \frac{1}{2} s(i|\mathcal{E}) \left(\frac{1 - (R_i^H)^r}{1 - R_i^H} - \frac{1 - (R_i^L)^r}{1 - R_i^L} \right) \\
= \ & \frac{1}{2} s(i|\mathcal{E}) \sum_{j=0}^{r-1} ((R_i^H)^j - (R_i^L)^j) \\
\leq \ & \frac{1}{2} s(i|\mathcal{E}) \sum_{j=0}^{\infty} ((R_i^H)^j - (R_i^L)^j) = \frac{1}{2} s(i|\mathcal{E}) \left(\frac{1}{1 - R_i^H} - \frac{1}{1 - R_i^L} \right)
\end{aligned}$$

$$
\begin{aligned}
&= \frac{s(i|\mathcal{E})(u_{max}(i|\mathcal{E}) - u_{min}(i|\mathcal{E}))}{2(s(i|\mathcal{E}) + u_{min}(i|\mathcal{E}))(s(i|\mathcal{E}) + u_{max}(i|\mathcal{E}))} \\
&\leq \frac{u_{max}(i|\mathcal{E}) - u_{min}(i|\mathcal{E})}{2\ s(i|\mathcal{E})}
\end{aligned}
$$

□

Theorem 6.

$$|\mathcal{F}_r^A(\mathcal{E}) - E[BP(G_0, G_r)]| \leq \sum_{i=0}^{k} \frac{u_{max}(i|\mathcal{E}) - u_{min}(i|\mathcal{E})}{2s(i|\mathcal{E})}, \ \ \forall k \geq 0$$

In addition, if $u_{max}(i|\mathcal{E})$ *(and thus* $u_{min}(i|\mathcal{E})$*) is* $O(s(i|\mathcal{E})/k)$, $\forall i : 0 \leq i \leq k$, *(the case for random inversions, transpositions, and inverted transpositions), then* $|\mathcal{F}_r^A(\mathcal{E}) - E[BP(G_0, G_r)]| = O(1)$.

Proof. The error is at most one half of the maximum difference between $\sum_{i=0}^{k} P_{i|r}^H(\mathcal{E})$ and $\sum_{i=0}^{k} P_{i|r}^L(\mathcal{E})$; the result follows from Lemma 2.

When both $u_{min}(i|\mathcal{E})$ and $u_{max}(i|\mathcal{E})$ are $O(\frac{s(i|\mathcal{E})}{n})$, the error is at most

$$\sum_{i=0}^{k} \frac{u_{max}(i|\mathcal{E}) - u_{min}(i|\mathcal{E})}{2\ s(i|\mathcal{E})} = \sum_{i=0}^{k} O(\frac{1}{k}) = O(1)$$

□

Theorem 7. *Let* $s_l = \min_{0 \leq i \leq k} \{s(i|\mathcal{E})\}$, $s_h = \max_{0 \leq i \leq k} \{s(i|\mathcal{E})\}$, $r_l = \min_{0 \leq i \leq k} \{s(i|\mathcal{E}) + u_{min}(i|\mathcal{E})\}$, *and* $r_h = \max_{0 \leq i \leq k} \{s(i|\mathcal{E}) + u_{max}(i|\mathcal{E})\}$. *For all* $r \geq 1$,

$$\frac{s_l r_l}{s_h r_h} \leq \frac{\mathcal{F}_r^A(\mathcal{E})}{E[BP(G_0, G_r)]} \leq \frac{s_h r_h}{s_l r_l}$$

In addition, if $s_h/s_l = 1 + \Theta(\frac{1}{k})$ *and* $u_{max}(i|\mathcal{E})$ *(and thus* $u_{min}(i|\mathcal{E})$*) is* $O(s(i|\mathcal{E})/k)$, $\forall i : 0 \leq i \leq k$, *then*

$$\frac{\mathcal{F}_r^A(\mathcal{E})}{E[BP(G_0, G_r)]} = 1 + O(\frac{1}{k})$$

Proof. We only prove the upper bound, as the lower bound is the reciprocal of the upper bound and can be proven similarly. Let $w = 1 - r_l$ and $v = 1 - r_h$; we have $v \leq w$, $1 - w^r \leq 1 - v^r$, and

$$\frac{\mathcal{F}_r^A(\mathcal{E})}{E[BP(G_0, G_r)]} = \frac{\sum_{i=0}^{k} P_{i|r}^H}{\sum_{i=0}^{k} P_{i|r}^L} \leq \frac{\max\limits_{0 \leq i \leq k} P_{i|r}^H}{\min\limits_{0 \leq i \leq k} P_{i|r}^L} \leq \frac{s_h(1 + w + w^2 + \ldots w^{r-1})}{s_l(1 + v + v^2 + \ldots v^{r-1})}$$

$$= \frac{s_h \dfrac{1 - w^r}{1 - w}}{s_l \dfrac{1 - v^r}{1 - v}} = \left(\frac{s_h(1 - v)}{s_l(1 - w)}\right)\left(\frac{1 - w^r}{1 - v^r}\right) \leq \frac{s_h(1 - v)}{s_l(1 - w)} = \frac{s_h r_h}{s_l r_l}$$

□

In Tables 3.1 and 3.2 are lists of the parameters of the three rearrangement classes in the GNT Model for linear and circular genomes.

3.3.6 Upper and lower bounds estimation with multiple rearrangement classes

We can easily extend the results to a mixture of different rearrangement classes. Consider m classes of rearrangements, $\mathcal{E}_1, \ldots, \mathcal{E}_m$, where $\mathcal{E}_i = (A(\mathcal{E}_i), f_{\mathcal{E}_i}), 1 \leq i \leq m$. For any rearrangement ρ, let $\gamma_j = \Pr(\rho \in \mathcal{E}_j), 1 \leq j \leq m$. Assume γ_j does not depend on genome G, the genome currently acted upon. Let $s(i|\mathcal{E}_j)$, $u(i|G, \mathcal{E}_j)$, $u_{min}(i|\mathcal{E}_j)$, and $u_{max}(i|\mathcal{E}_j)$ be the parameters corresponding to $\mathcal{E}_j$ as defined in Theorem 5. Let $\mathcal{E} = (A(\mathcal{E}), f_{\mathcal{E}})$ be the rearrangement class such that $A(\mathcal{E}) = \cup_{j=1}^{m} A(\mathcal{E}_j)$, and $f_{\mathcal{E}}(r|G) = \sum_{j=1}^{m} \gamma_j \ f_{\mathcal{E}_j}(r|G)$. Then $\text{Sep}(i|G, \mathcal{E}) = \cup_{j=1}^{m} \text{Sep}(i|G, \mathcal{E}_j)$, and $\text{Uni}(i|G, \mathcal{E}) = \cup_{j=1}^{m} \text{Uni}(i|G, \mathcal{E}_j)$.

The hierarchical way of choosing rearrangements (first choose rearrangement class, then choose one rearrangement among others in the class chosen) during evolution allows two rearrangements in different rearrangement classes to produce the same results, while retaining the additivity of

Linear Genomes							
Signed	Rearrangement type	$s(i)$ $i \neq 0, k$	s_0, s_k	$u_{min}(i)$ $i \neq 0, k$	$u_{max}(i)$ $i \neq 0, k$	$u_{min}(0)$ $u_{min}(k)$	$u_{max}(0)$ $u_{max}(k)$
No	Inv	$\frac{k-2}{\binom{k}{2}}$	$\frac{2}{k}$	$\frac{2}{\binom{k}{2}}$	$\frac{2}{\binom{k}{2}}$	$\frac{1}{\binom{k}{2}}$	$\frac{1}{\binom{k}{2}}$
Yes	Inv	$\frac{2}{k+1}$	$\frac{2}{k+1}$	0	$\frac{1}{\binom{k+1}{2}}$	0	$\frac{1}{\binom{k+1}{2}}$
No	Trp	$\frac{3(k-2)}{k(k-1)}$	$\frac{3}{k+1}$	$\frac{6}{k(k-1)}$	$\frac{6}{k(k-1)}$	$\frac{1}{\binom{k+1}{3}}$	$\frac{6}{k(k+1)}$
Yes	Trp	$\frac{3}{k+1}$	$\frac{3}{k+1}$	0	$\frac{6}{k(k+1)}$	0	$\frac{6}{k(k+1)}$
No	Trv	$\frac{3(k-3)}{k(k-1)}$	$\frac{3}{k}$	$\frac{6}{k(k-2)}$	$\frac{6}{k(k-2)}$	$\frac{1}{2\binom{k}{3}}$	$\frac{6}{k(k-1)}$
Yes	Trv	$\frac{3}{k+1}$	$\frac{3}{k+1}$	0	$\frac{3}{(k-1)(k+1)}$	0	$\frac{3}{k(k+1)}$

Table 3.1: Recurrence parameters in the GNT model for linear genomes. The number of genes is k. Included are the two different data forms: signed or unsigned. The three rearrangement classes are inversion (Inv), transposition (Trp), and inverted transposition (Trv).

Circular Genomes				
Signed	Rearrangement type	$s(i)$ $1 \leq i \leq k$	$u_{min}(i)$ $1 \leq i \leq k$	$u_{max}(i)$ $1 \leq i \leq k$
No	Inv	$\frac{k-3}{\binom{k-1}{2}}$	$\frac{2}{\binom{k-1}{2}}$	$\frac{2}{\binom{k-1}{2}}$
Yes	Inv	$\frac{2}{k}$	0	$\frac{1}{\binom{k}{2}}$
No	Trp	$\frac{3(k-3)}{(k-1)(k-2)}$	$\frac{6}{(k-1)(k-2)}$	$\frac{6}{(k-1)(k-2)}$
Yes	Trp	$\frac{3}{k}$	0	$\frac{6}{k(k-1)}$
No	Trv	$\frac{3}{k-1}$	$\frac{6}{(k-1)(k-3)}$	$\frac{6}{(k-1)(k-3)}$
Yes	Trv	$\frac{3}{k}$	0	$\frac{4}{k(k-2)}$

Table 3.2: Recurrence parameters in the GNT model for circular genomes. The number of genes is k. Included are the two different data forms: signed or unsigned. The three rearrangement classes are inversion (Inv), transposition (Trp), and inverted transposition (Trv). Here $B_i(G_r)$ has the same distribution for $1 \leq i \leq k$, and $B_0(G_r)$ is always set to 0.

probability:

$$\begin{aligned} & \Pr(\rho = \rho_0 | G = G_0) \\ = & \sum_{j=1}^{m} \Pr(\rho = \rho_0 | G = G_0, \mathcal{E}_j \text{ is chosen}) \Pr(\mathcal{E}_j \text{ is chosen} | G = G_0) \\ = & \sum_{j=1}^{m} \gamma_j f_{\mathcal{E}_j}(\rho_0 | G_0) \end{aligned}$$

The new recurrence is

$$\begin{aligned} s(i|\mathcal{E}) &= \Pr(B_i(G_{r+1}) = 1 | B_i(G_r) = 0) \\ &= \Pr(\rho_r \in \text{Sep}(i|G_r, \mathcal{E}) \mid B_i(G_r) = 0) \\ &= \sum_{j=1}^{m} \Pr(\rho_r \in \text{Sep}(i|G_r, \mathcal{E}_j) \mid B_i(G_r) = 0) \\ &= \sum_{j=1}^{m} \gamma_j s(i|\mathcal{E}_j) \end{aligned}$$

Similarly,

$$\begin{aligned} u(i|G_r, \mathcal{E}) &= \Pr(B_i(G_{r+1}) = 0 | B_i(G_r) = 1) = \sum_{j=1}^{m} \gamma_j u_j(i|G_r, \mathcal{E}_j), \forall r \geq 0 \\ u_{min}(i|\mathcal{E}) &= \sum_{j=1}^{m} \gamma_j u_{min}(i|\mathcal{E}_j), \quad u_{max}(i|\mathcal{E}) = \sum_{j=1}^{m} \gamma_j u_{max}(i|\mathcal{E}_j) \\ P_{i|r+1} &= (1 - P_{i|r}) s(i|\mathcal{E}) + P_{i|r}(1 - u(i|G_r, \mathcal{E})) \\ &= P_{i|r}(1 - s(i|\mathcal{E}) - u(i|G_r, \mathcal{E})) + s(i|\mathcal{E}) \\ P_{i|0} &= 0 \end{aligned}$$

Results similar to Theorems 6 and 7 on error bounds can be obtained for multiple classes:

Theorem 8. *Consider the estimator $\mathcal{F}_r^A(\mathcal{E})$ defined in Definition 1 with the parameters $s(i|\mathcal{E})$, $u_{min}(i|\mathcal{E})$, and $u_{max}(i|\mathcal{E})$ in the previous paragraphs. If the*

assumptions in Theorems 6 and 7 regarding these parameters are satisfied, then

$$|\mathcal{F}_r^A(\mathcal{E}) - E[BP(G_0, G_r)]| = O(1)$$

and

$$\phi^{-1} \leq \frac{\mathcal{F}_r^A(\mathcal{E})}{E[BP(G_0, G_r)]} \leq \phi$$

where $\phi = 1 + O(\frac{1}{k})$.

Proof. Follows from Theorems 6 and 7. □

3.3.7 `Approx-IEBP` under the Generalized Nadeau-Taylor model

Recall that in the GNT model, all three types of rearrangements can occur: inversions, transpositions, and inverted transpositions. Given as part of the model are two values α and β such that the probability an rearrangement is an inversion, a transposition, or an inverted transpositions are $1 - \alpha - \beta$, α, and β, respectively. In this section we use the techniques given above in order to derive a *t.e.d.* estimator between genomes, when the permitted rearrangements include inversions, transpositions, and inverted transpositions, and given arbitrarily defined probabilities on the three classes of rearrangements. We use $s(i|\alpha, \beta)$ and $u(i|\alpha, \beta)$ to denote the corresponding s and u parameters.

In Tables 3.3 and 3.4 are the parameters for the upper-lower bounds technique.

Recall the estimator given in Definition 1. Under the GNT model we can tighten the error bounds obtained in Theorem 6, as follows:

Signed	Genome	$s(i)\ \ (1 \leq i \leq k-1)$	$u_{min}(i)\ \ (1 \leq i \leq k-1)$	$u_{max}(i)\ \ (1 \leq i \leq k-1)$
No	Lin	$\frac{2(k-2)+\alpha(k-2)+\beta(k-5)}{k(k-1)}$	$\frac{2(2k-4)+2\alpha(k-2)+2\beta(k+1)}{k(k-1)(k-2)}$	$\frac{2(2k-4)+2\alpha(k-2)+2\beta(k+1)}{k(k-1)(k-2)}$
Yes	Lin	$\frac{2+\alpha+\beta}{k+1}$	0	$\frac{2(k-1)+4\alpha(k-1)+\beta(k+2)}{(k+1)k(k-1)}$
No	Cir	$\frac{2(k-3)+\alpha(k-3)+\beta k}{(k-1)(k-2)}$	$\frac{4(k-3)+2\alpha(k-3)+2\beta k}{(k-1)(k-2)(k-3)}$	$\frac{4(k-3)+2\alpha(k-3)+2\beta k}{(k-1)(k-2)(k-3)}$
Yes	Cir	$\frac{2+\alpha+\beta}{k}$	0	$\frac{2(k-2)+4\alpha(k-2)+2\beta k}{k(k-1)(k-2)}$

Signed	Genome	$s(0), s(k)$	$u_{min}(0), u_{min}(k)$	$u_{max}(k), u_{max}(k)$
No	Lin	$\frac{2(k+1)+\alpha(k-2)+\beta(k+1)}{k(k+1)}$	$\frac{2(k+1)(k-2)-2\alpha(k-2)^2}{(k+1)k(k-1)(k-2)}$ $-\frac{\beta(k+1)(2k-7)}{(k+1)k(k-1)(k-2)}$	$\frac{2(k+1)+4\alpha(k-2)+4\beta(k+1)}{(k+1)k(k-1)}$
Yes	Lin	$\frac{2+\alpha+\beta}{k+1}$	0	$\frac{2+4\alpha+\beta}{k(k+1)}$

Table 3.3: Recurrence parameters for the GNT model. The probability a rearrangement is an inversion, a transposition, or an inverted transposition, are $1-\alpha-\beta$, α, and β, respectively. For circular genomes, the parameters $s(k)$, $u_{min}(k)$, and $u_{max}(k)$ agree with those for $i = 1, \ldots, k-1$, and $B_0(G) = 0$ for all genomes G.

Sign	Genome	Absolute error bound $= \sum_{i=0}^{k} \frac{u_{max}(i)-u_{min}(i)}{2s(i)}$	Relative error upper bound† $= \frac{\sum_{i=0}^{k} P^H_{i\|r}}{\sum_{i=0}^{k} P^L_{i\|r}}$
No	Lin	$\frac{6\alpha+\beta\left(2+\frac{9(k-1)}{(k-2)^2}\right)}{(k-1)\left(2+\alpha+\beta+\frac{3(1+\beta)}{k-2}\right)} \leq \frac{2}{k-1}$	$1+O(k^{-2})$
Yes	Lin	$\frac{1}{2}+\frac{3\alpha}{2(2+\alpha+\beta)}+\frac{1}{k}\left(2-\frac{3}{2+\alpha+\beta}\right) \leq 1+\frac{1}{k}$	$1+\frac{2+4\alpha+\beta}{2+\alpha+\beta}k^{-1}+O(k^{-2})$
No	Cir	0	1
Yes	Cir	$\left(1+\frac{3\alpha+\beta\left(\frac{k+2}{k-2}\right)}{2+\alpha+\beta}\right)\left(\frac{k}{2(k-1)}\right) \leq 1+\frac{1}{k-1}$	$1+\frac{2+4\alpha+2\beta}{2+\alpha+\beta}k^{-1}+O(k^{-2})$

† See Theorems 7 and 9 for details. Only the upper bounds are shown here; the lower bounds are their reciprocals.

Table 3.4: Error bounds for the GNT model in the Approx-IEBP distance. The probability a rearrangement is an inversion, a transposition, or an inverted transposition, are $1-\alpha-\beta$, α, and β, respectively. For circular genomes, the parameters $s(k)$, $u_{min}(k)$, and $u_{max}(k)$ agree with those for $i = 1, \ldots, k-1$, and $B_0(G) = 0$ for all genomes G.

Theorem 9. *We assume the genomes evolve under the GNT model. For all* $r > 0$,

$$|\mathcal{F}_r^A(\alpha, \beta) - E[BP(G_0, G_r)]| \leq 1 + \frac{1}{k-1}$$

and

$$\phi^{-1} \leq \frac{\mathcal{F}_r^A(\alpha, \beta)}{E[BP(G_0, G_r)]} \leq \phi$$

where $\phi = \frac{2+4\alpha+2\beta}{2+\alpha+\beta} k^{-1} + O(k^{-2})$.

Proof. Follows from Theorems 6 and 7 with parameters $s(i|\alpha, \beta)$, $u_{min}(i|\alpha, \beta)$ and $u_{max}(i|\alpha, \beta)$. See Tables 3.3 and 3.4 for details. For the relative error bound, we look at $\frac{\sum_{i=0}^{k} P_{i|r}^H}{\sum_{i=1}^{k} P_{i|r}^L}$ directly to improve the result. □

We now define the `Approx-IEBP` estimator:

Algorithm `Approx-IEBP`

Input: Two genomes G and G' having the same set of distinct genes, GNT model parameter α and β.

Output: Integer $r = \widehat{r}^A(G, G')$, an estimate of the true evolutionary distance between G and G'.

1. Compute the breakpoint distance b between G and G'.
2. Compute $s(i|\alpha, \beta)$, $u_{min}(i|\alpha, \beta)$ and $u_{max}(i|\alpha, \beta)$, required in $\mathcal{F}_r^A(\alpha, \beta)$.
3. Return integer r such that $\left|\mathcal{F}_r^A(\alpha, \beta) - b\right|$ is minimized.

3.3.8 Running time analysis

Similar to the analysis of `Exact-IEBP` (see Section 3.2), the value q, the number of inversions needed to produce a genome that is close to random, is an upper bound of r. We can improve the running time by the following implementation. For each k, the value k that minimizes $|\mathcal{F}_r^A - D_{BP}(G)|$ can be found in $O(\log q) = O(\log k)$ time using the bisection method. Since there are at most $k+1$ distinct nonzero breakpoint distance values, we create an array that stores the `Approx-IEBP` distances corresponding to each possible breakpoint distance value, and use the corresponding breakpoint distance values as indices. When a new breakpoint distance value is encountered we compute the `Approx-IEBP` distance and store it in the array. We summarize the discussion as follows:

Theorem 10. *Let k be the number of genes in each genome, and let n be the number of genomes. We can compute the `Approx-IEBP` distances of all $\binom{n}{2}$ pairs of genomes in $O(n^2k + \min\{k, n^2\}\log k)$ time.*

3.4 The EDE distance estimator

Although NJ using our IEBP estimators show marked improvement over NJ using breakpoint or inversion distances, it too degrades in accuracy when given data close to saturation. This degradation motivated us to design a correction function to apply to input distance matrices so as to improve the behavior of NJ on nearly saturated data. We used extensive simulations to obtain large amounts of information on the relationship between actual and minimal distances, then designed a correction function, EDE, using various fitting tools and numerical techniques.

To develop an estimator for the actual number of inversions under which NJ performs well, we simulated evolution for a large range of numbers k of genes and numbers r of (random) inversions. We then normalized observed values by the number of genes, plotted the (normalized) actual number of inversions against the (normalized) minimum inversion distance, computed the means of the sets of values for which x is fixed, and graphed this mean. The curve we obtained suggested a function F mapping normalized numbers of actual inversions to normalized inversion distances. This function F must have the following properties:

1. $0 \leq F(x) \leq x$ (obviously).
2. $\lim_{x\to\infty} F(x) = a_k$, where a_k is the expected inversion distance between two random genomes on k genes, divided by k.
3. $F'(0) = 1$, because initially every inversion increases the inversion distance by 1.
4. $F^{-1}(y)$ is defined for all $y \in [0, 1]$. We also assume that F is monotone increasing (additional inversions generally, if not always, increase the inversion distance) to allow us to infer F^{-1}.

A ratio of second-degree polynomials satisfies constraints (2)–(4), so we used $F_0(x) = \frac{ax^2+bx}{x^2+cx+b}$.

Experiments showed that setting $a = 1$ for all values of k produces the best results. To estimate b and c, we minimized the least-square error between F_0 and the empirical data—that is, we minimized $\sum_{(x,y)} |F_0(x) - y|^2$. Using gradient descent methods, we obtained $b = 0.5956$ and $c = 0.4577$. Because this definition of F_0 does not always satisfy constraint (1), we set $F(x) = \min\{F_0(x), x\}$; this is the "fixed-point modification."

We can now define EDE to be the nonnegative inverse of F. EDE overestimates the actual number of inversions for large inversion distances. However, this overestimation appears not to affect the performance of NJ (we explored several ways of modifying the latter values, but did not obtain an improvement).

Chapter 4

Estimating the Variances of Genomic Distances

4.1 Introduction

Variance of genomic distances The following problem has been studied in Chapter 3: given any genome G with n genes, what is the expected breakpoint distance between G and G' when G' is the genome obtained from G by applying k rearrangements according to the GNT model? Our approach is to compute the probability of having a breakpoint between every pair of genes; by linearity of the expectation the expected breakpoint distance can be obtained by n times the aforementioned probability. Each breakpoint can be characterized as a Markov process with $2(n-1)$ states. But the probability of a breakpoint is a sum of $O(n)$ terms that we do not know yet how to further simplify.

However the variance cannot be obtained this way since breakpoints are not independent (under any evolutionary model) by the following simple observation: the probability of having a breakpoint for each breakpoint position is nonzero, but the probability of the breakpoint distance being 1 is zero (the breakpoint distance is always 0 or at least 2). Thus, to compute the variance (or the second moment) of the breakpoint distance we need to look at two breakpoints at the same time. This implies we have to study a Markov

[1]The content of this chapter also appeared in [84].

process of $O(n^2)$ states and a sum of $O(n^2)$ terms that is hard to simplify. As for the inversion distance, even the expectation is still an open problem.

Estimating the variance of breakpoint and inversion distances is important for several reasons. Based on these estimates we can compute the variances of the `Approx-IEBP` and `Exact-IEBP` estimators (based on the breakpoint distance), and the `EDE` estimator (based on the inversion distance). It is also informative when we compare estimators based on breakpoint distances to other estimators, e.g. the inversion distance and the `EDE` distance. Finally, variance estimation can be used in distance-based methods to improve the topological accuracy of tree reconstruction.

Outline of this chapter We start in Section 4.2 by presenting a stochastic model approximating the breakpoint distance, and derive the analytical form of the variance of the approximation, as well as the variance of the `IEBP` estimators. In Section 4.3 the variances of the inversion and the `EDE` distances are obtained through simulation. Based on these variance estimates we propose four new methods, called `BioNJ-IEBP`, `Weighbor-IEBP`, `BioNJ-EDE`, and `Weighbor-EDE`. These methods are based on `BioNJ` and `Weighbor`, but the variances in these algorithms have been replaced with the variances of `IEBP` and `EDE`.

4.2 Variance of the Breakpoint and `IEBP` Distances

The approximating model We first define the following notation: $\binom{a}{b}$ is the number of ways of choosing b objects from a (the binomial coefficient) when $a \geq b \geq 0$; $\binom{a}{b}$ is set to 0 otherwise.

We motivate the approximating model by the case of inversion-only evolution on signed circular genomes. Let n be the number of genes, and b be the number of breakpoints of the current genome G. When we apply a random inversion (out of $\binom{n}{2}$ possible choices) to G, we have the following cases according to the two endpoints of the inversion [28]:

1. None of the two endpoints of the inversion is a breakpoint. The number of breakpoints is increased by 2. There are $\binom{n-b}{2}$ such inversions.

2. Exactly one of the two endpoints of the inversion is a breakpoint. The number of breakpoints is increased by 1. There are $b(n-b)$ such inversions.

3. The two endpoints of the inversion are two breakpoints. There are $\binom{b}{2}$ such inversions. Let g_i and g_{i+1} be the left and right genes at the left breakpoint, and let g_j and g_{j+1} be the left and right genes at the right breakpoint. There are three subcases:

   (a) None of $(g_i, -g_j)$ and $(-g_{i+1}, g_{j+1})$ is an adjacency in G_0. The number of breakpoints is unchanged.

   (b) Exactly one of $(g_i, -g_j)$ and $(-g_{i+1}, g_{j+1})$ is an adjacency in G_0. The number of breakpoints is decreased by 1.

   (c) $(g_i, -g_j)$ and $(-g_{i+1}, g_{j+1})$ are adjacencies in G_0. The number of breakpoints is decreased by 2.

When $b \geq 3$, out of the $\binom{b}{2}$ inversions from case 3, case 3(b) and 3(c) count for at most b inversions; this means given that an inversion belongs to case 3, with probability at least $1 - b/\binom{b}{2} = \frac{b-3}{b-1}$ it does not change the

breakpoint distance; this probability is close to 1 when b is large. Furthermore, when $b << n$ almost all the inversions belong to case 1 and 2. Therefore, when n is large, we can drop cases 3(b) and 3(c) without affecting the distribution of breakpoint distance drastically.

The approximating model we use is as follows. Assume first the evolutionary model is such that each rearrangement creates r breakpoints on an unrearranged genome (for example, $r = 2$ for inversions and $r = 3$ for transpositions and inverted transpositions). Let us be given n boxes, initially empty. At each iteration r boxes will be chosen randomly (without replacement); we then place a ball into each of these r boxes if it is empty. The number of nonempty boxes after k iterations, b_k, can be used to estimate the number of breakpoints after k rearrangement events are applied to an unrearranged genome. This model can also be extended to approximate the GNT model: at each iteration, with probability $1 - \alpha - \beta$ we choose 2 boxes, and with probability $\alpha + \beta$ we choose 3 boxes.

Mean and variance of the approximating model Fix n (the number of boxes) and k (the number of times we choose r boxes). Consider the expansion of the following expression

$$S = ((x_1x_2 + x_1x_3 + \cdots + x_{n-1}x_n)/\tbinom{n}{2})^k$$

Each term corresponds to the number of ways of choosing $r = 2$ boxes for k times where the total number of times box i is chosen is the power of x_i, and the coefficient of that term is the total probability of these ways. For example, the coefficient of $x_1^3x_2x_3^2$ in S (when $k = 3$) is the probability of choosing box 1 three times, box 2 once, and box 3 twice. Let u_i be the sum of the coefficients

of those terms with i distinct symbols; $\binom{n}{i}u_i$ is the probability i boxes are nonempty after k iterations. The identity of u_i for all terms of the same set of power indices holds as long as the probability of each box being chosen is identical; in other words, S is not changed by permuting $\{x_1, x_2, \ldots, x_n\}$ arbitrarily.

To solve for u_i exactly for all k is difficult and unnecessary. Instead we can find the expectation and variance of b_k directly. Actually the following results give all moments of b_k. Let $S(a_1, a_2, \ldots, a_n)$ be the value of S when we substitute x_i by a_i, $1 \le i \le n$, and let S_j be the value of S when $a_1 = a_2 = \cdots = a_j = 1$ and $a_{j+1} = a_{j+2} = \cdots = a_n = 0$. For integers j, $0 \le j \le n$, we have

$$\sum_{i=0}^{j} \binom{j}{i} u_i = S(\underbrace{1, 1, 1, \ldots, 1}_{j \ 1's}, 0, \ldots, 0) = S_j$$

Let

$$\begin{aligned} Z_a &= \sum_{i=0}^{n} i(i-1)\cdots(i-a+1)\binom{n}{i}u_i \\ &= \sum_{i=a}^{n} n(n-1)\cdots(n-a+1)\binom{n-a}{i-a}u_i \end{aligned}$$

for all a, $1 \le a \le n$. We want to express Z_a by some linear combination of S_i, $0 \le i \le n$. The following lemma, which is a special case of equation (5.24) in [27], finds the coefficients of the linear combination.

Lemma 3. *Let a be some given integer such that $1 \le a \le n$. Let us be given $\{u_i : 0 \le i \le n\}$ that satisfy $\sum_{j=0}^{i}\binom{i}{j}u_j = \sum_{j=0}^{n}\binom{i}{j}u_j = S_i$,where $0 \le i \le n$. We have $\sum_{i=n-a}^{n}(-1)^{n-i}\binom{a}{n-i}S_i = \sum_{j=0}^{n}\binom{n-a}{j-a}u_j$.*

The following results follow from Lemma 3; the proofs are straghtforward.

Theorem 11. *For all a, $1 \leq a \leq n$,*

$$Z_a = n(n-1)\cdots(n-a+1)\sum_{i=n-a}^{n}(-1)^{n-i}\binom{a}{n-i}S_i.$$

Corollary 3. *(a)* $Eb_k = Z_1 = n(1-S_{n-1})$.

(b) $Var\ b_k = nS_{n-1} - n^2S_{n-1}^2 + n(n-1)S_{n-2}$.

These results work for all integers r, $1 \leq r \leq n$. When there are more than one type of rearrangement events with different r's we can change S accordingly. For example, let $\gamma = \alpha + \beta$; for the GNT model we can set

$$S = \left(\frac{1-\gamma}{\binom{n}{2}}\Big(\sum_{1\leq i_1<i_2\leq n} x_{i_1}x_{i_2}\Big) + \frac{\gamma}{\binom{n}{3}}\Big(\sum_{1\leq i_1<i_2<i_3\leq n} x_{i_1}x_{i_2}x_{i_3}\Big)\right)^k. \tag{4.1}$$

Mean and variance of the breakpoint distance under the GNT model
We begin this section by finding the mean and variance for b_k with respect to the GNT model. By substituting into equation (4.1):

$$S_{n-1} = (1-\frac{2+\gamma}{n})^k, S_{n-2} = \left(\frac{(n-3)(n-2-2\gamma)}{n(n-1)}\right)^k$$

For the GNT model, we have the following results:

$$\begin{aligned}
\frac{d}{dk}Eb_k &= -nS_{n-1}(\frac{1}{k}\ln S_{n-1}) = -nS_{n-1}\ln(1-\frac{2+\gamma}{n}) && (4.2)\\
\text{Var } b_k &= nS_{n-1} - n^2S_{n-1}^2 + n(n-1)S_{n-2} \\
&= (nS_{n-1} - n^2S_{n-1}^2 + n(n-1)S_{n-2}) && (4.3)
\end{aligned}$$

Using `BioNJ` *and* `Weighbor`. Both `BioNJ` and `Weighbor` are designed for DNA sequence phylogeny using the variance of the true evolutionary distance estimator. In `BioNJ`, the distance update step of NJ is modified so the variances of the new distances are minimized. In `Weighbor`, the pairing step is also modified to utilize the variance information. We use the variance for the GNT model in this section and the expected breakpoint distance in [83] in the two methods. The new methods are called `BioNJ-IEBP` and `Weighbor-IEBP`.

To estimate the true evolutionary distance, we use `Exact-IEBP`, though we can also use Equation (4.2) which is less accurate. Let $\widehat{k}(b)$ denote the `Exact-IEBP` distance given the breakpoint distance is b; $\widehat{k}(b)$ behaves as the inverse of Eb_k, the expected breakpoint distance after k rearrangement events. The variance of $\widehat{k}(b)$ can be approximated using a common statistical technique, the delta method [57], as follows:

$$\text{Var } \widehat{k}(b) \simeq (\frac{d}{dk}Eb_k)^{-2}\text{Var } b_k = \frac{\left(1 - nS_{n-1} + (n-1)(\frac{S_{n-2}}{S_{n-1}})\right)}{nS_{n-1}(\ln(1 - \frac{2+\gamma}{n}))^2}$$

When the number of rearrangements are below the number of genes (120 in the simulation), these results are accurate approximations to the mean and variance of the breakpoint distance under the GNT model. Also the approximation is less accurate when transpositions and inverted transpositions are present.

As the number of rearrangements k is so high the breakpoint distance is close to the maximum (the resulting genome is random with respect to the genome before evolution), the simulation shows the variance is much lower than the theoretical formula. This is due to the application of the delta method: while the method assumes the `Exact-IEBP` distance is continuous, in reality it is a discrete function. The effect gets more obvious as k is large: different values

of k all give breakpoint distances close to the maximum, yet the `Exact-IEBP` can only return one estimate for k, hence the very low variance. This problem is less serious as n increases.

4.3 Variance of the Inversion and `EDE` Distances

The `EDE` distance Given two genomes having the same set of n genes and the inversion distance between them is d, we define the `EDE` distance as $nf^{-1}(\frac{d}{n})$: here n is the number of genes, and f, an approximation to the expected inversion distance normalized by the number of genes, is defined as (see Chapter 3):

$$f(x) = \min\{x, \frac{ax^2 + bx}{x^2 + cx + b}\}$$

We simulate the inversion-only GNT model to evaluate the relationship between the inversion distance and the actual number of inversion applied. Regression on simulation results suggests $a = 1$, $b = 0.5956$, and $c = 0.4577$. As the rational function is inverted, we take the larger (and only positive) root:

$$x = \frac{-(b - cy) \pm \sqrt{(b - cy)^2 + 4(a - y)by}}{2(a - y)}$$

Let $y = \frac{d}{n}$. Thus

$$f^{-1}(y) = \max\{y, \frac{-(b - cy) \pm \sqrt{(b - cy)^2 + 4(a - y)by}}{2(a - y)}\}$$

Here the coefficients do not depend on n, since for different values of n the curves of the normalized expected inversion distance are similar.

Regression for the Variance Due to the success of nonlinear regression in the derivation of `EDE`, we use the same technique again for the variance

of the inversion distance (and that of EDE). However for different numbers of genes, the curves of the variance are very different (see Figure 4.1). From the simulation it is obvious the magnitudes of the curves are inversely proportional to the number of genes (or some kind of function of it).

We use the following regression formula for the standard deviation of the inversion distance normalized by the number of genes after nx inversions are applied:

$$g_n(x) = n^q \frac{ux^2 + vx}{x^2 + wx + t}$$

The constant term in the numerator is zero because we know $g(0) = 0$. Let r be the value such that rn is the largest number of inversions applied; we use $r = 2.5$. Note that

$$\ln(\frac{1}{rn}\sum_{0}^{rn} g_n(x)) \simeq \ln(\frac{1}{r}\int_0^r g_n(x)dx) = q\ln n + \ln(\frac{1}{r}\int_0^r \frac{ux^2 + vx}{x^2 + wx + t}dx)$$

is a linear function of $\ln n$. Therefore, we can obtain q as the slope in the linear regression using n as the independent variable and $\ln(\frac{1}{rn}\sum_0^{rn} g_n(x))$ as the independent variable (see Figure 4.1(b); simulation results suggest the average of the curve indeed is inversely proportional to $\ln n$). When q is obtained we apply nonlinear regression to obtain u, v, w, and t using the simulation data for 40, 80, 120, and 160 genes. The resultant functions are shown as the solid curves in Figure 4.1, with coefficients $q = -0.6998$, $u = 0.1684$, $v = 0.1573$, $w = -1.3893$, and $t = 0.8224$.

Variance of the EDE Distance Let X_k and Y_k be the inversion and EDE distances after k inversions are applied to a genome of n genes, respectively.

We again use the delta method. Let $x = \frac{k}{n}$. Since $X_k = nf(\frac{Y_k}{n})$, we have

$$\left|\frac{dY_k}{dX_k}\right|^{-1} = \left|\frac{dX_k}{dY_k}\right| = \frac{1}{n}\left|\frac{dX_k}{d(Y_k/n)}\right| = f'(x) = \frac{d}{dx}\left(\min\{x, \frac{x^2+bx}{x^2+cx+b}\}\right)$$

The point where $x = \frac{x^2+bx}{x^2+cx+b}$ is when $x = 0.5423$. Therefore

$$f'(x) = \begin{cases} 1 & \text{if } 0 \le x < 0.5423 \\ \frac{d}{dx}(\frac{x^2+bx}{x^2+cx+b}) = \frac{x^2(c-b)+2bx+b^2}{(x^2+cx+b)^2} & \text{if } x \ge 0.5423 \end{cases}$$

and $Var(Y_k) \simeq \left|\frac{dY_k}{dX_k}\right|^2 Var(X_k) = (f'(x))^{-2}(ng_n(x))^2 = (ng_n(\frac{k}{n})/f'(\frac{k}{n}))^2$.

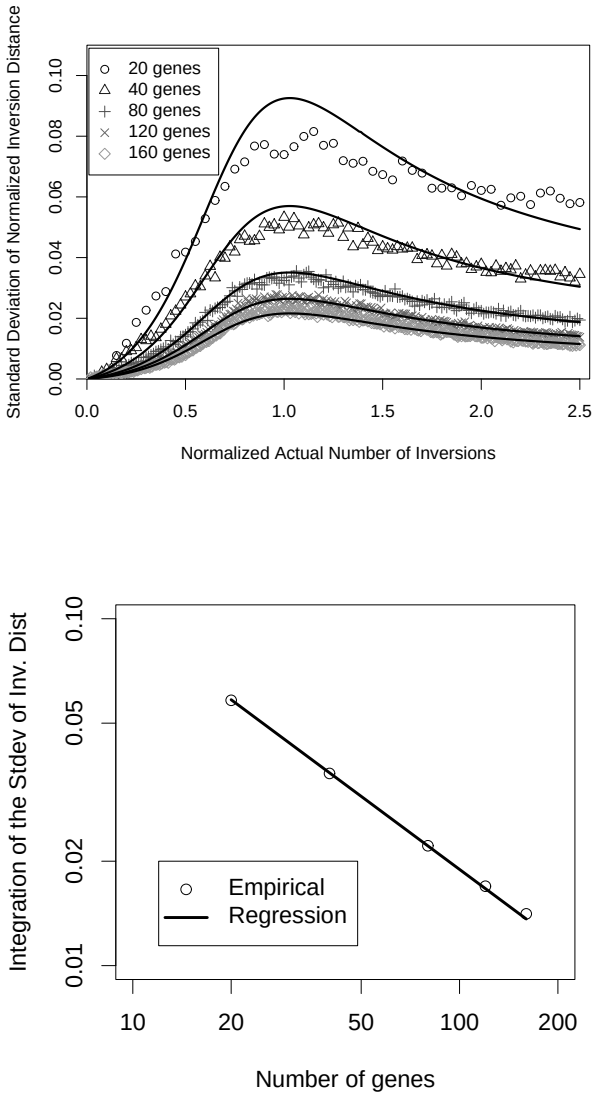


Figure 4.1: *Top: simulation (points) and regression (solid lines) of the standard deviation of the inversion distance.* Bottom: regression of coefficient q (see Section 4.3); for every point corresponding to n genes, the y coordinate is the average of all data points in the simulation.

Chapter 5

Simulation Studies of Distance-based Genome Rearrangement Phylogeny Methods

In this chapter we present the results of the simulation studies comparing the different true evolutionary distance estimators and estimates of their variances, as well distance-based genome rearrangement phylogeny reconstruction methods. Their derivations are presented in the previous two chapters.

5.1 The accuracies of the true evolutionary distance estimators

In this section we study the behavior of the `Exact-IEBP`, `Approx-IEBP`, and `EDE` distances by comparing them to the actual number of rearrangement events. We simulate the GNT model on a circular genome with 37 (the typical number of genes in the animal mitochondrial genomes [9]), and 120 genes (the typical number of genes in the plant chloroplast genomes [35]). Starting with the unrearranged genome G_0, we apply r events to it to obtain the genome G_r, where $r = 1, \ldots, 300$. For each value of r we simulate 500 runs. We then compute the five distances.

The simulation results under the inversion-only model are shown in Figures 5.1, 5.2, and 5.3. Under the other two model settings, the simulation

[1]The content of this chapter also appeared in [50, 83, 84, 86].

results show similar behavior (e.g. shape of curves and standard deviations). Note that both BP and INV distances underestimate the actual number of events. The slope of both curves are higher when the model allows transpositions and inverted transpositions. When the number of genes increases the standard deviations of both methods decrease.

We then compare different distance estimators by the absolute difference in the measured distances and the actual number of events. Using the same data in the previous experiment, we generate the plots as follows. The x-axis is the actual number of events. For each distance estimator D we plot the curve f_D, where $f_D(x)$ is the mean of the set $\{|\frac{1}{c}D(G_0, G_r) - r| : 1 \leq r \leq x\}$ over all observations G_r.[1]

The result is in Figure 5.4. The relative performance is the same for most cases: BP is the worst, followed by INV, and Approx-IEBP. Exact-IEBP has the best performance. In most cases, Approx-IEBP has similar behavior as Exact-IEBP when the amount of evolution is small; the Approx-IEBP and Exact-IEBP curves are almost indistinguishable in (a). Yet, in other figures the Approx-IEBP curve is inferior than the Exact-IEBP curve by a large margin when the number of events is above about 200. When there are more transpositions and inverted transpositions, the gap between Approx-IEBP and Exact-IEBP becomes larger; this effect is stronger when the number of genes is 37. In the extreme case, when the number of genes is 37 Approx-IEBP can

[1]The constant c is to reduce the bias effect in different distances. For the Approx-IEBP and the Exact-IEBP distances $c = 1$ since they estimate the actual number of events. For the BP distance we let $c = 2(1 - \alpha - \beta) + 3(\alpha + \beta) = 2 + \alpha + \beta$ since this is the expected number of breakpoints created by each event in the model when the number of events is very low. Similarly for the INV distance we let $c = (1 - \alpha - \beta) + 3\alpha + 2\beta = 1 + 2\alpha + \beta$ since each transposition can be replaced by 3 inversions, and each inverted transposition can be replaced by 2 inversions.

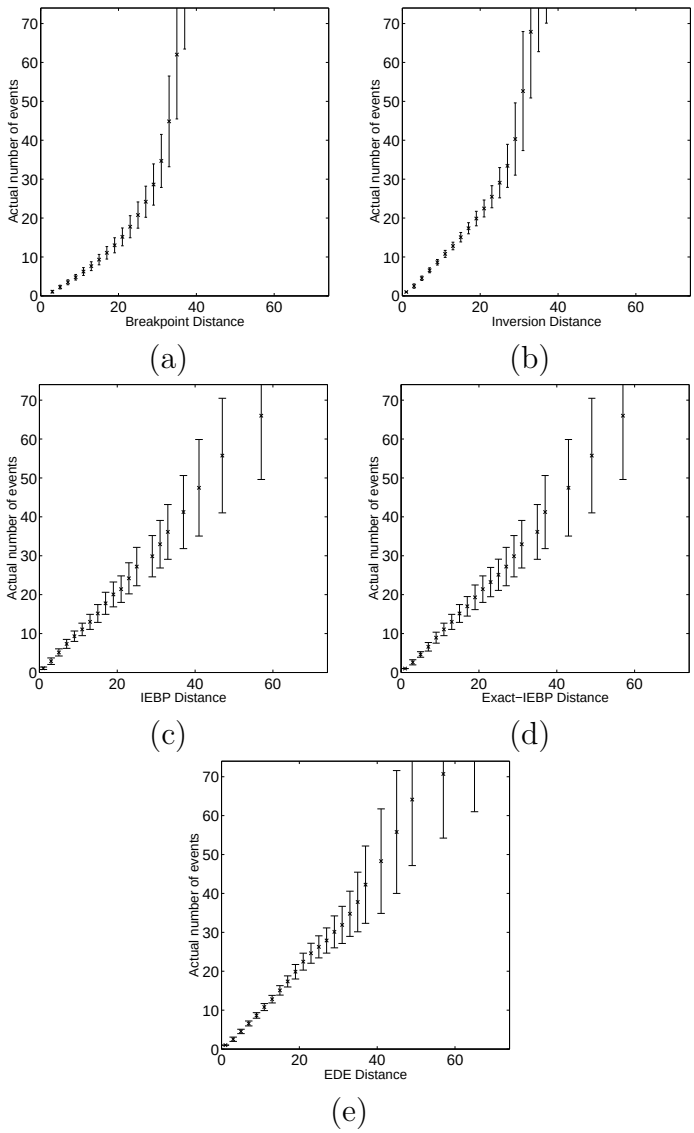


Figure 5.1: Accuracies of the estimators (see Section 5.1). The number of genes is 37. See Section 5.1 for more details. The evolutionary model is inversion-only. The x-axis is divided into 25 bins; the length of the vertical bars indicate the standard deviation. The distance estimators are (a) `BP`, (b) `INV`, (c) `Approx-IEBP`, (d) `Exact-IEBP`, and (e) `EDE`.

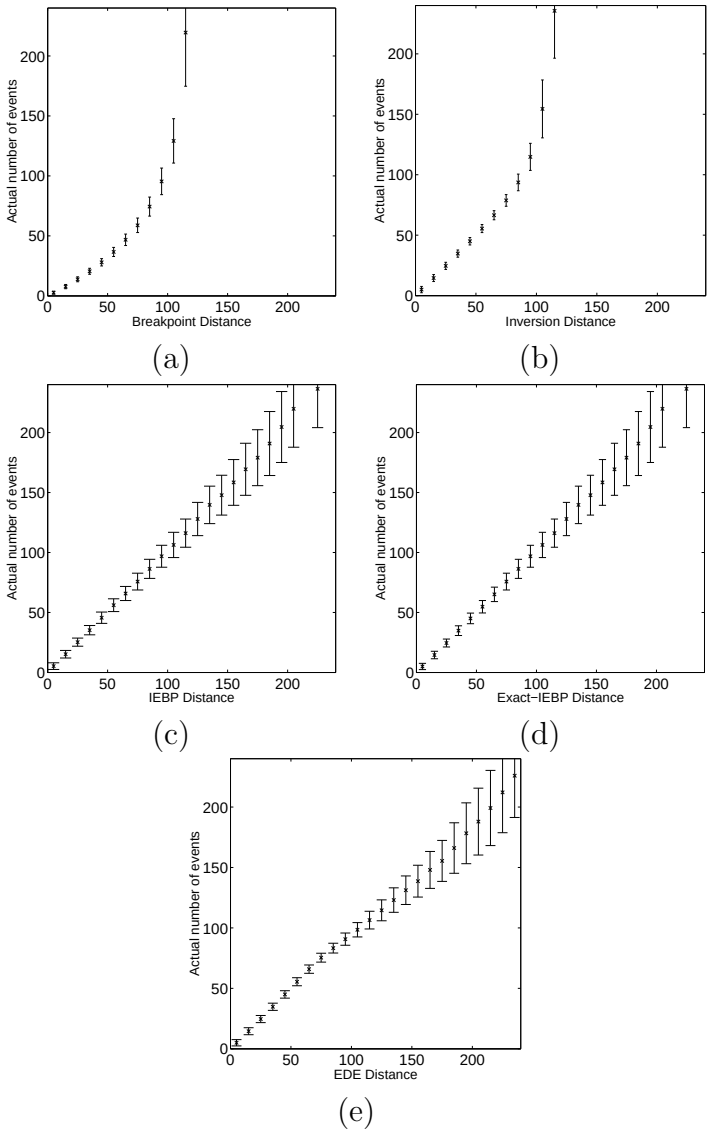


Figure 5.2: Accuracies of the estimators (see Section 5.1). The number of genes is 120. See Section 5.1 for more details. The evolutionary model is inversion-only. The x-axis is divided into 25 bins; the length of the vertical bars indicate the standard deviation. The distance estimators are (a) `BP`, (b) `INV`, (c) `Approx-IEBP`, (d) `Exact-IEBP`, and (e) `EDE`.

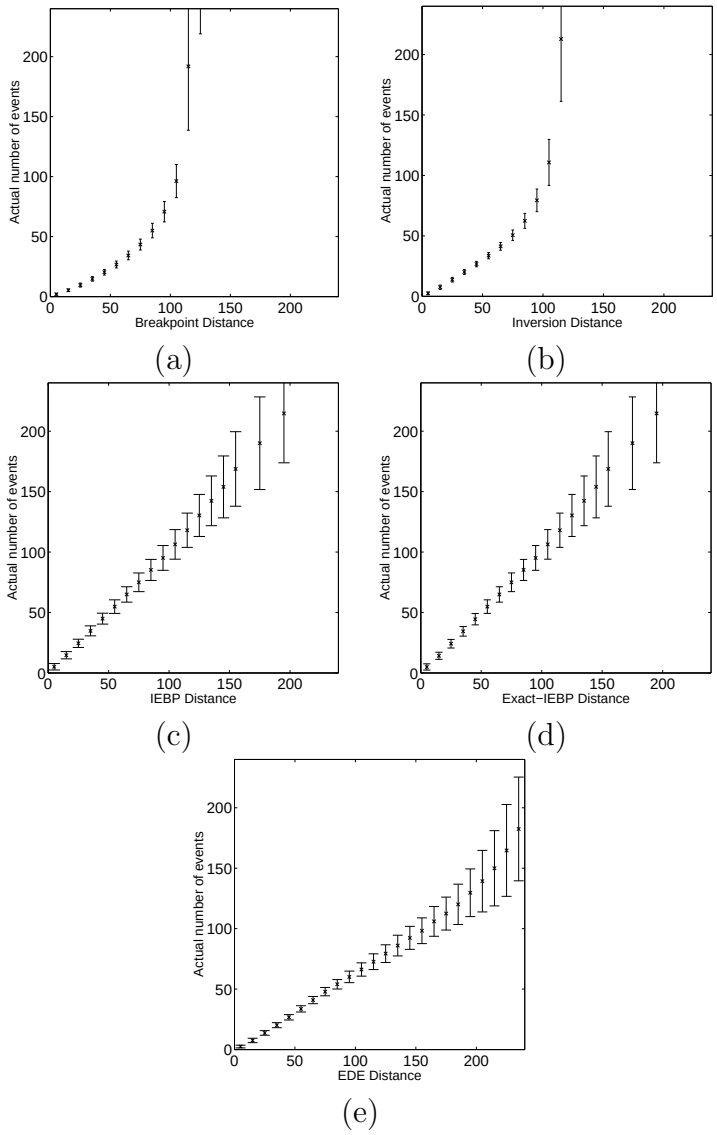


Figure 5.3: Accuracies of the estimators (see Section 5.1). The number of genes is 120. See Section 5.1 for more details. The evolutionary model is such that the three types of rearrangement events are equiprobable ($\alpha = \beta = 1/3$). The x-axis is divided into 25 bins; the length of the vertical bars indicate the standard deviation. The distance estimators are (a) `BP`, (b) `INV`, (c) `Approx-IEBP`, (d) `Exact-IEBP`, and (e) `EDE`.

be the worst when the model is transposition-only.

5.2 The accuracies of the variance estimates of true evolutionary distance estimators

5.2.1 The variances of `BP` and `Exact-IEBP`

We present the result of simulations for the accuracies of the variance estimates of `BP` and `Exact-IEBP`, the derivations of which can be found in Chapter 4. The setting of the simulation is the same as that for the true evolutionary distance estimators: we simulate the GNT model on a circular genome with 120 genes. Starting with the unrearranged genome G_0, we apply r events to it to obtain the genome G_r, where $r = 1, \ldots, 300$. For each value of r we simulate 500 runs. We then compute the five distances.

When the number of rearrangements are below the number of genes (120 in the simulation), these results are accurate approximations to the mean and variance of the breakpoint distance under the GNT model. Also the approximation is less accurate when transpositions and inverted transpositions are present.

As the number of rearrangements k is so high the breakpoint distance is close to the maximum (the resulting genome is random with respect to the genome before evolution), the simulation shows the variance is much lower than the theoretical formula. This is due to the application of the delta method: while the method assumes the `Exact-IEBP` distance is continuous, in reality it is a discrete function. The effect gets more obvious as k is large: different values of k all give breakpoint distances close to the maximum, yet the `Exact-IEBP` can only return one estimate for k, hence the very low variance. This problem is less serious as n increases.

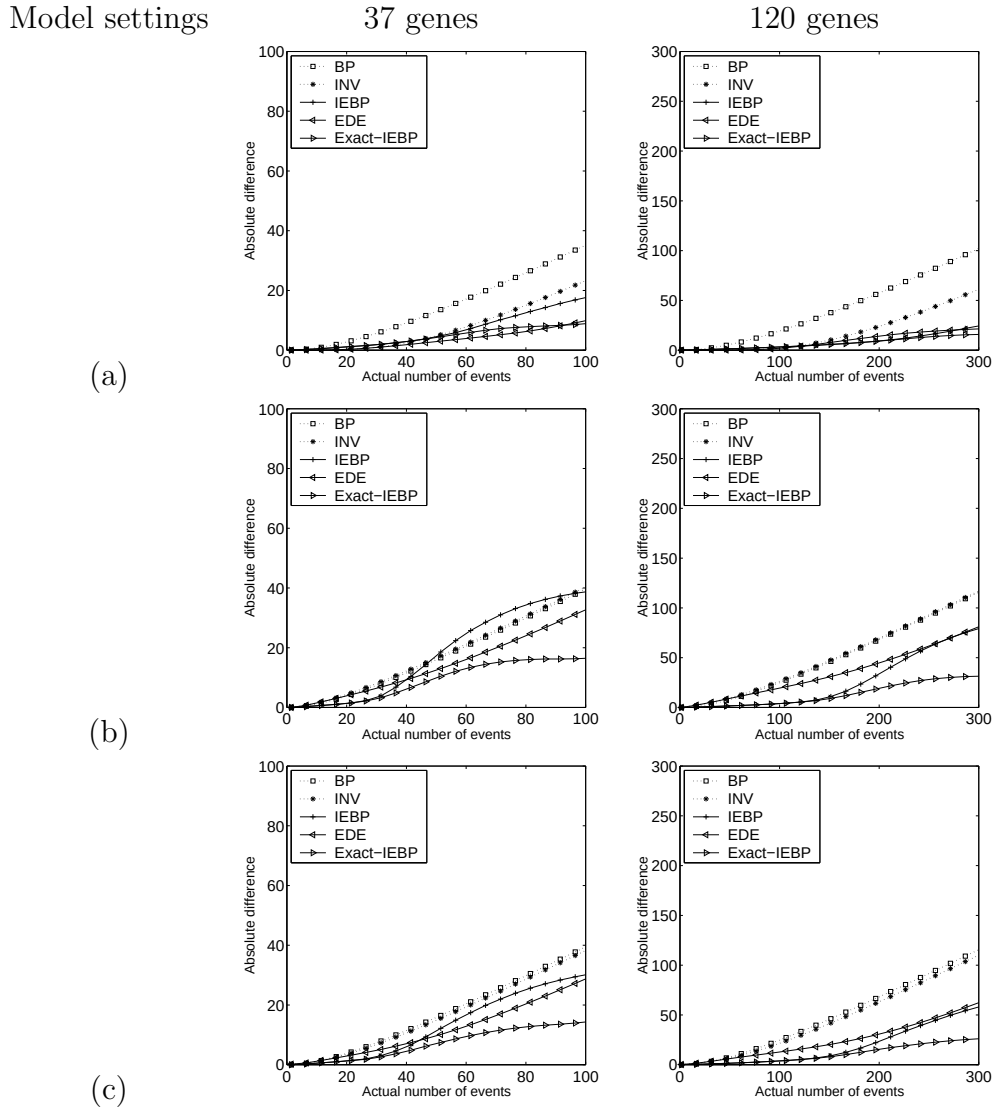


Figure 5.4: Accuracies of the estimators by absolute difference (See Section 5.1 for the details). We simulate the evolution on 37 and 120 genes. The evolutionary models are (a) Inversions only, (b) Transpositions only, (c) Three types of events equally likely.

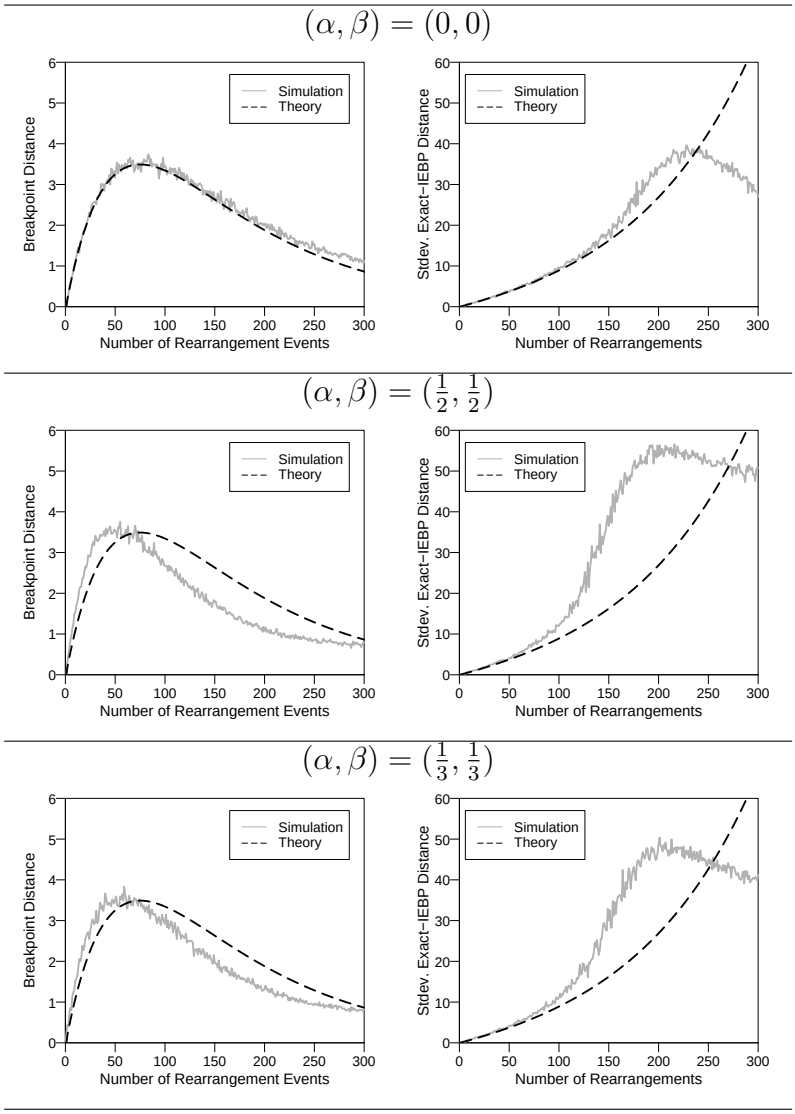


Figure 5.5: Accuracies of the estimator for the variance. Each figure consists of two sets of curves, corresponding to the values of simulation and theoretical estimation. The number of genes is 120. The number of rearrangement events, k, range from 1 to 300. The evolutionary model is inversion-only GNT. For each k we generate 500 runs. We then compute the standard deviation of b_k for each k, and those of $\widehat{k}(b_k)$ for each k, and compare them with the values of the theoretical estimation. See definition of $\widehat{k}$ in page 66.

5.2.2 The variances of INV and EDE

Simulation results of the variance estimates and INV and EDE are shown in Figure 4.1 in Chapter 4, as a part of the derivation.

5.3 The accuracies of distance-based tree reconstruction methods

In this section we present the comparison of the accuracies of our new methods with other distance-based methods for genome rearrangement phylogeny.

5.3.1 Settings

We use the following methods for tree reconstructions:

1. The following four distance estimators are used with neighbor joining,
   (a) BP, the breakpoint distance,
   (b) INV, the inversion distance,
   (c) Exact-IEBP, and
   (d) EDE, true evolutionary distance estimators based on BP and INV, respectively.
2. The following four methods use the variance of the true evolutionary distance estimator they use:
   (a) BioNJ-IEBP and BioNJ-EDE,
   (b) Weighbor-IEBP and Weighbor-EDE.

Table 5.1: Settings for Experiments on the accuracies of tree reconstructions.

Parameter	Value
1. Number of genes	120 (plant chloroplast genome)
2. Model tree generation	Uniformly Random Topology
(See the Model Tree paragraph in Section 5.3 for details.)	
4. GNT Model parameters $(\alpha, \beta)^\dagger$	$(0, 0)$, $(\frac{1}{4}, \frac{1}{4})$
5. Datasets for each setting	30

† The probabilities that a rearrangement is an inversion, a transposition, or an inverted transposition are $1 - \alpha - \beta$, α, and β, respectively.

Their definitions can be found in previous chapters. The procedure of neighbor joining combined with distance `X` will be denoted by `NJ(X)`. See Table 5.1 for the settings for the experiment.

Quantifying error Given an inferred tree, we compare its "topological accuracy" by computing "false negatives" with respect to the "true tree", which is defined on page 28; it is defined as the percentage of internal edges in the true tree that are false negative edges with respect to the inferred tree. Please see more details on the true tree and false negative rates in Chapter 2.

Software We use PAUP* 4.0 [78] to compute the neighbor joining method and the false negative rates between two trees. We have implemented a simulator [50, 86] for the GNT model. The input is a rooted leaf-labeled model tree $(T, \{\lambda_e\})$, and parameters (α, β). On each edge, the simulator applies random rearrangement events to the circular genome at the ancestral node according to the model with given parameters α and β. We use the original `Weighbor` and `BioNJ` implementations [11, 25] (downloadable from the internet) and make modifications so they use the new variance formulas.

Model Trees The model trees have topologies drawn from the uniform distribution [2], and edge lengths drawn from the discrete uniform distribution on intervals $[1, b]$, where b is one of the following: 3, 6, 12, 18 (higher values of b makes the variance of the edge lengths higher). Then the length of each edge is scaled by the same factor so the diameter of the tree (the maximum pairwise leaf-to-leaf distance on the tree) is 36, 72, 120, 180, 360 (so low- to high-evolutionary rates are covered).

5.3.2 Results

The simulation results are plotted in Figures 5.6, 5.7, and 5.8. For each setting for simulation, we group methods based on the genomic distances they are based on: breakpoint or inversion distance.

In subsequent paragraphs, we present our observations by presenting subsets of the whole simulation and reorganize the curves in more accessible ways, depending on the points we want to make. The curves for `NJ(Approx-IEBP)` have been removed to make these figures more readable; in fact the first point we make in the discussion (along with figures specially made for that point) is that `NJ(Approx-IEBP)` and `NJ(Exact-IEBP)` have very similar accuracy curves, and `NJ(Exact-IEBP)` is always better. However, other comparisons on the whole experiments can still be made directly from Figures 5.6, 5.7, and 5.8.

Discussions We make the following observations:

[2] This is easily done and well known by the community by adding one leaf at a time to produce the whole tree. At each iteration, we choose an edge from the current tree (each edge has the same probability to be chosen) and attach the new leaf to it.

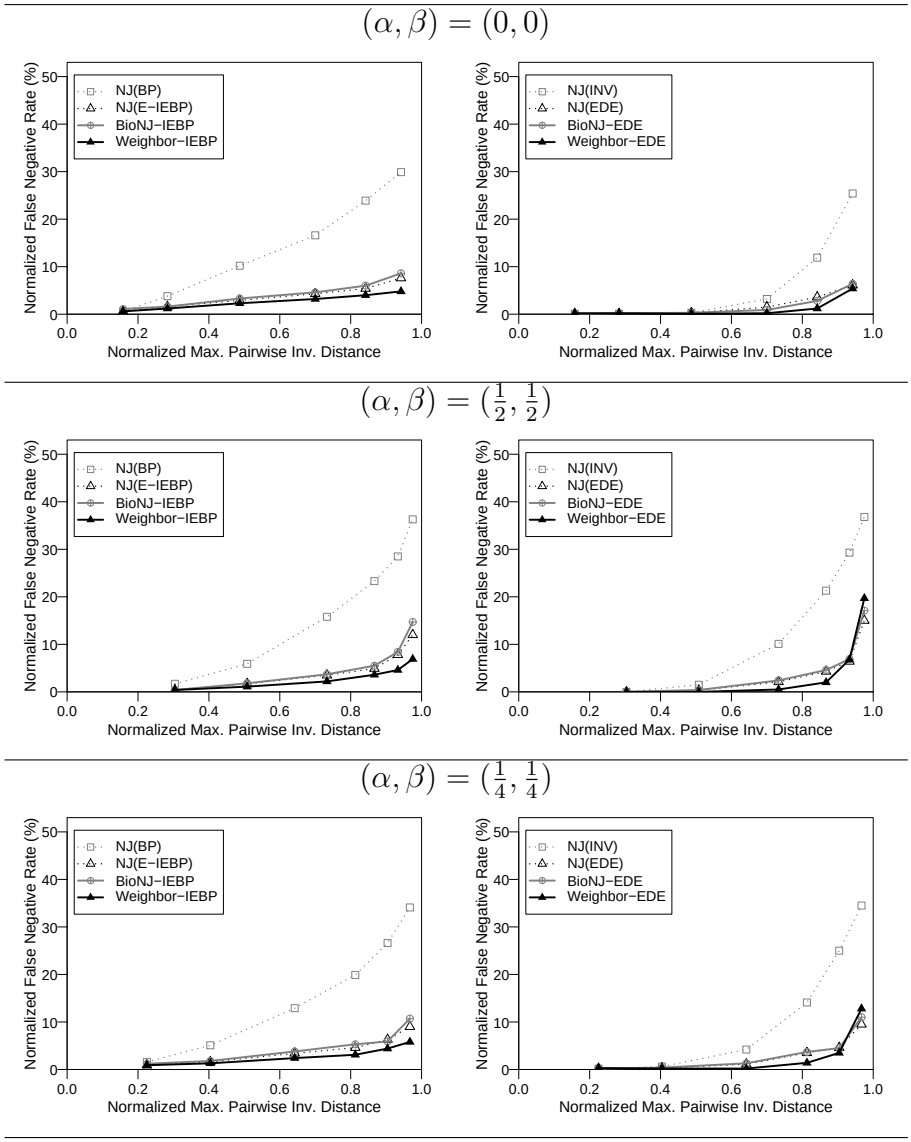


Figure 5.6: The topological accuracies of various distance-based tree reconstruction methods. The number of genes is 120 and the number of genomes is 40.

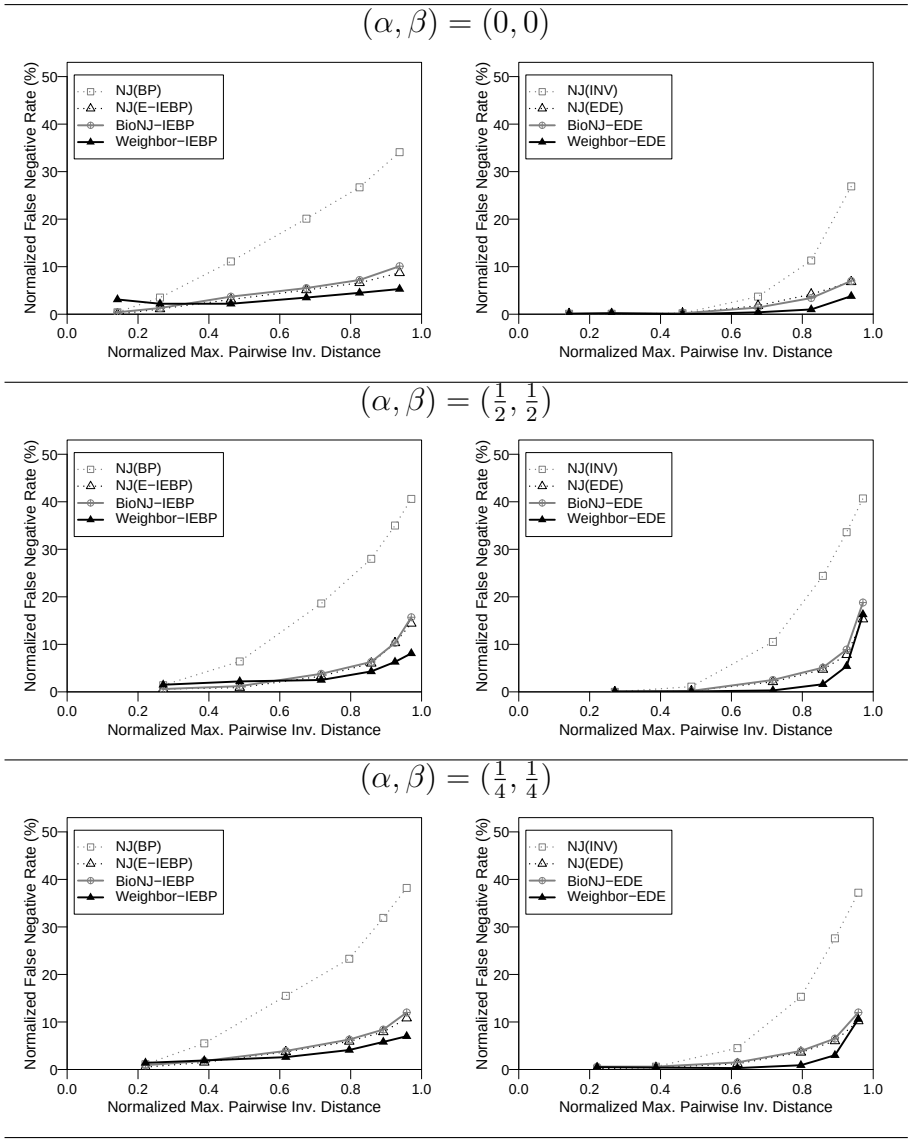


Figure 5.7: The topological accuracies of various distance-based tree reconstruction methods. The number of genes is 120 and the number of genomes is 80.

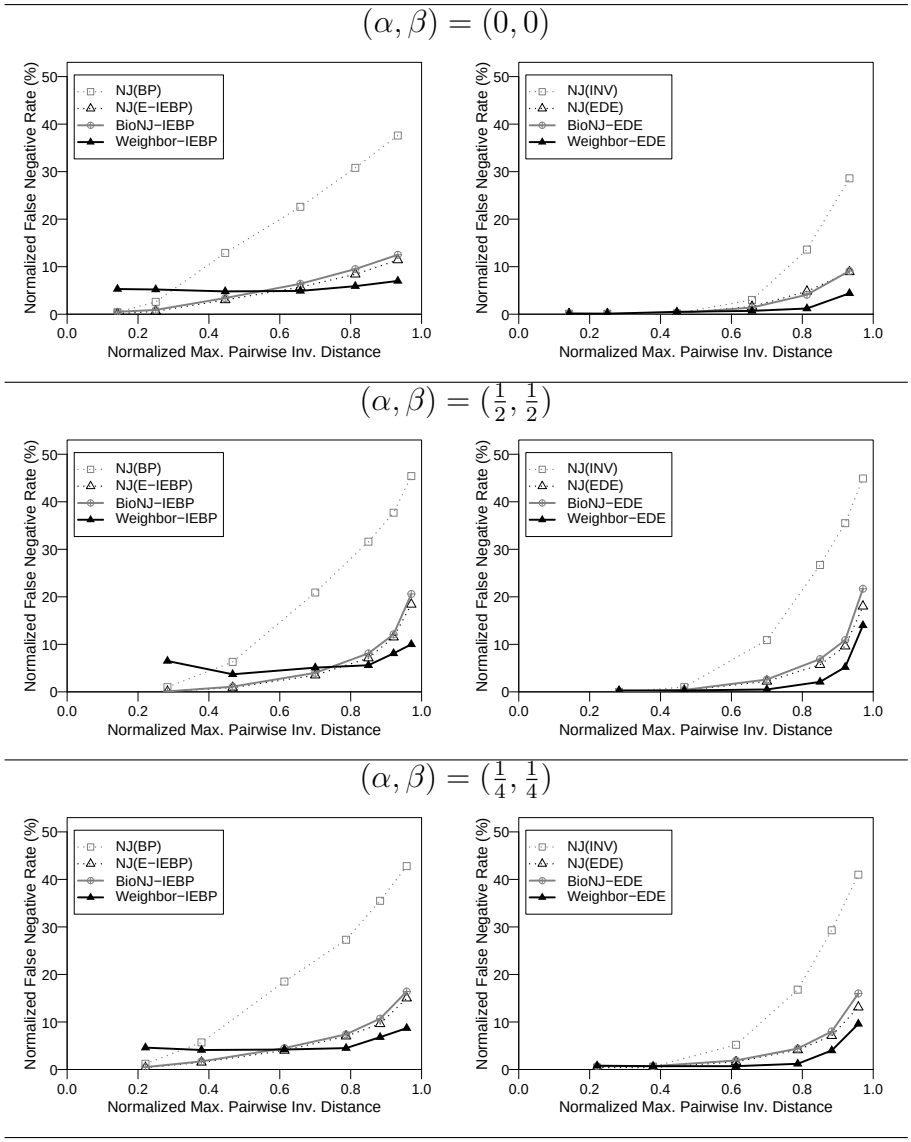


Figure 5.8: The topological accuracies of various distance-based tree reconstruction methods. The number of genes is 120 and the number of genomes is 160.

1. *NJ(Approx-IEBP) and NJ(Exact-IEBP) have similar accuracy.*

   See Figure 5.9 for the comparison. In particular, NJ(Exact-IEBP) is slightly better than NJ(Approx-IEBP) when the maximum pairwise inversion distance (i.e. the evolutionary rate) of the dataset is high.

2. *Corrected distances are better than uncorrected distances, and Weighbor is better than NJ. Furthermore, Weighbor-EDE has the best accuracy over all methods.*

   The comparison can be made directly from Figures 5.6, 5.7, and 5.8. In these figures, the relative order of accuracy is roughly the same in either the BP-based group (left) or the INV-based group (right). Let Y be the true distance estimator based on a genomic distance X; e.g. Y=IEBP if X=BP, and Y=EDE if X=INV. The order of the methods, starting from the worst, is (1) NJ(X), (2) BioNJ-Y, (3) NJ(Y), and (4) Weighbor-Y (except for very low evolutionary rates when Weighbor-IEBP is worst, but only by a few percents). The differences between NJ(Y) and BioNJ-Y are extremely small.

   Note Weighbor-IEBP outperforms NJ(Exact-IEBP) when the normalized maximum pairwise inversion distance, or the diameter of the dataset, exceeds 0.6; Weighbor-IEBP (based on BP) is even better than NJ(EDE) (based on the better INV) when the diameter of the dataset exceeds 0.9. This suggests the Weighbor approach really shines under high amounts of evolution.

3. *Inversion distance is better than breakpoint distance.*

   See Figures 5.10 and 5.11 for the comparison of a subset of the experimental results.

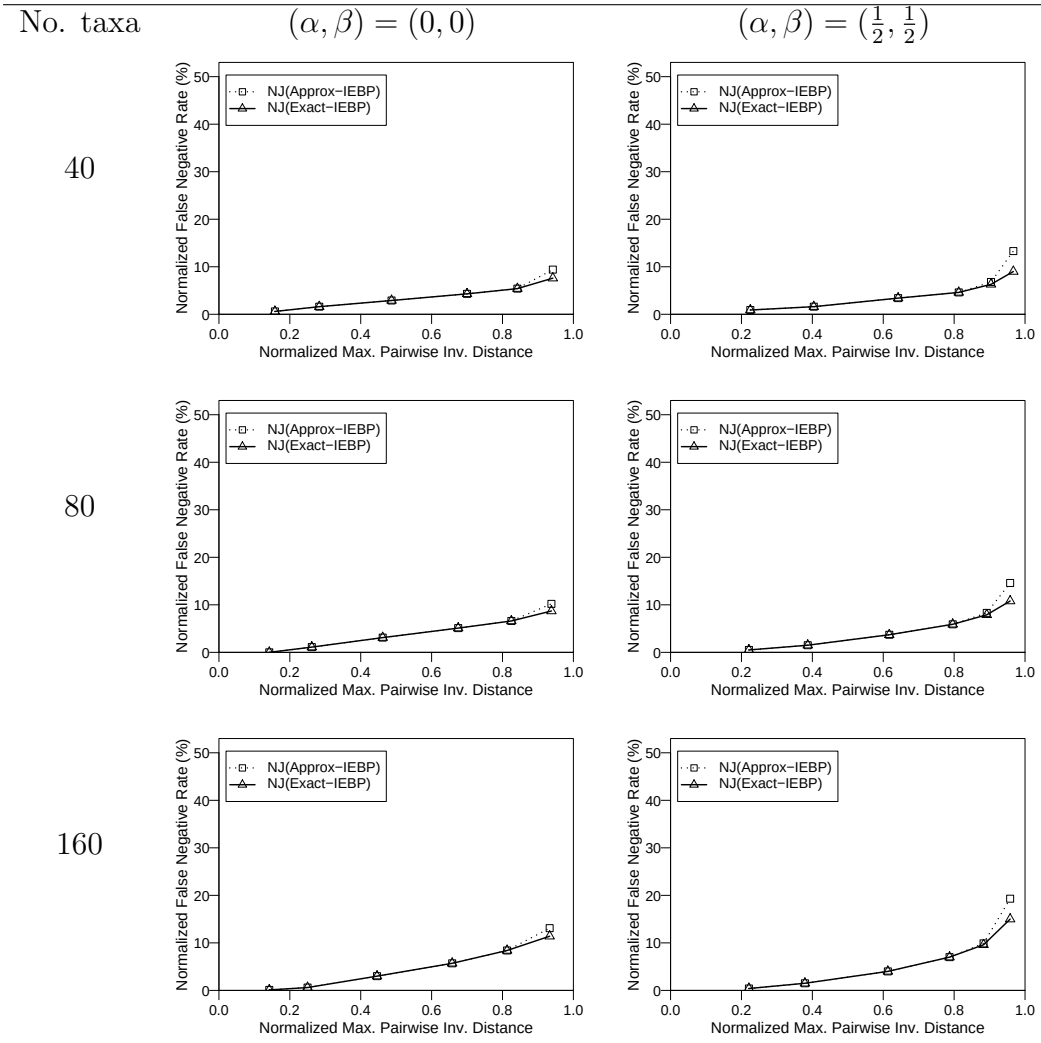


Figure 5.9: Comparison of `NJ(Exact-IEBP)` and `NJ(Approx-IEBP)`. The number of genes is 120.

When we compare methods based on breakpoint distance and methods based on inversion distance, the latter are always better (or no worse) than the former if we compare methods of the same complexity: `NJ(INV)` is better than `NJ(BP)`, `NJ(EDE)` is better than `NJ(Exact-IEBP)`, `BioNJ-EDE` is better than `BioNJ-IEBP` (shown in Figures 5.6, 5.7, and 5.8), and `Weighbor-EDE` is better than `Weighbor-IEBP` (with very small exceptions when transpositions and inverted transpositions are presented and the maximum pairwise inversion distance is hight).

This suggests INV is a better statistic than BP for the true evolutionary distance under the GNT model, even when transpositions and inverted transpositions are present. This is not surprising as INV, just like BP, increases by a small constant when a rearrangement event from the GNT model is applied. Also, though their maximum allowed values are the same (the number of genes for circular signed genomes), the fact the average increase in INV is smaller[3] than the average increase in BP gives INV a wider effective range.

4. *Increase in the number of taxa hurts the accuracy of reconstructed trees.*

   See Figures 5.12 and 5.13. Generally speaking, increase in the number of taxa means drop in the accuracy, though the effect is smaller for certain cases such as `Weighbor-EDE` or lower rates of evolution. Notice the way

[3] An inversion creates two breakpoints; a transposition and an inverted transposition can be realized by three and two inversions, respectively, and they all create three breakpoints each. Thus under the GNT model with model parameters (α, β) and assumption that genome G has only a small breakpoint ($BP(G, G_0)$) and inversion ($INV(G, G_0)$) distance from the reference (ancestral) genome G_0, the average increase in $BP(G, G_0)$ after a random rearrangement is applied to G is $2(1-\alpha-\beta)+3\alpha+3\beta = 2+\alpha+\beta$ and the average increase in $INV(G, G_0)$ is $(1-\alpha-\beta)+3\alpha+2\beta = 1+2\alpha+\beta$. The latter is always smaller, and the two quantities are equal only when $\alpha = 1$, i.e. only transpositions occur.

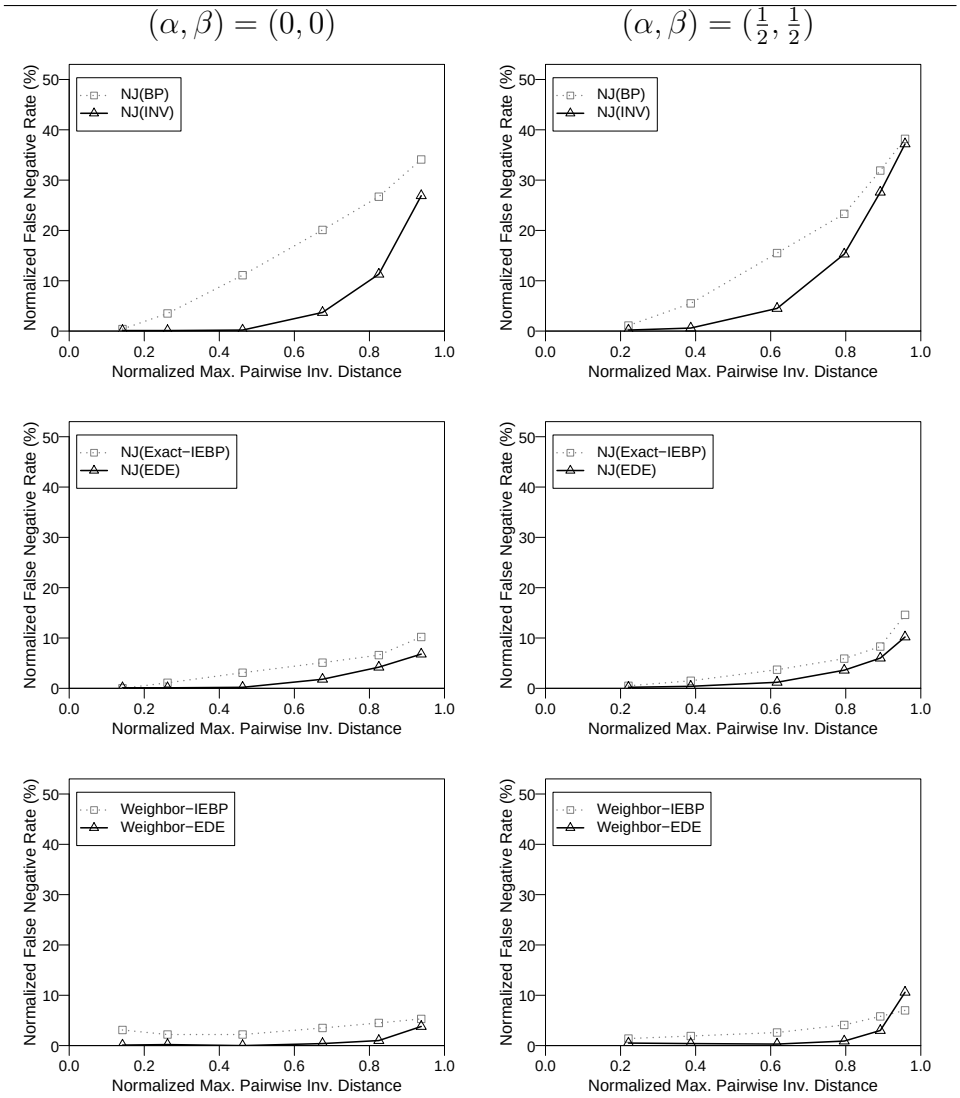


Figure 5.10: Comparing methods based on BP to methods based on INV. The number of genes is 120 and the number of taxa is 80.

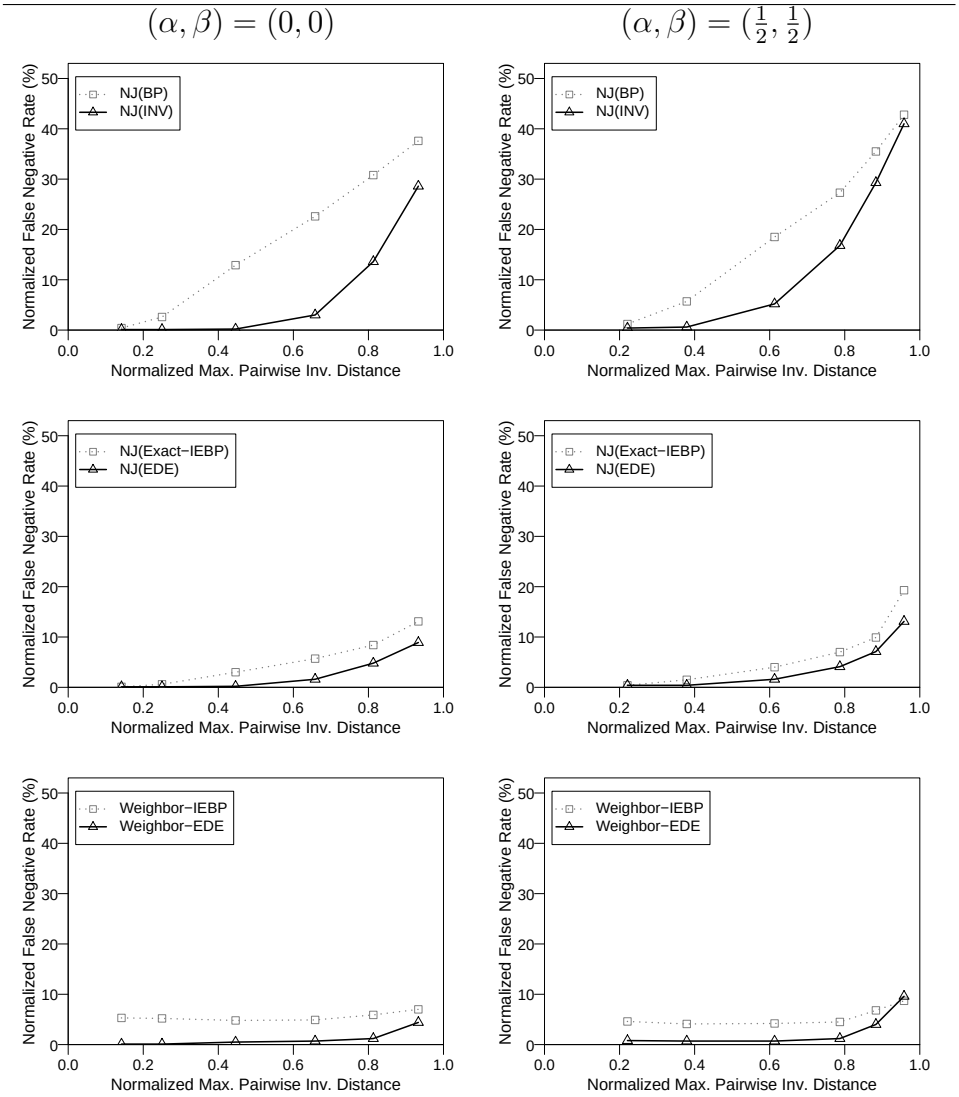


Figure 5.11: Comparing methods based on BP to methods based on INV. The number of genes is 120 and the number of taxa is 160.

we generate the model tree: by scaling every edge in the tree by a same constant, we can control the diameter of the tree, and hence the maximum pairwise inversion distance; in other words, the number of taxa has little effect in the evolutionary rate of the whole dataset.

5. *Increase in the number of genes helps the accuracy of reconstructed trees.*

   See Figure 5.14 for the results. Here we use the result from [83]. In addition to the number of genes we use (37 genes, the typical number of genes in animal mitochondria, and 120 genes as before), the setting of the experiment is slightly different. First, the trees are generated in a different way: we simply set the edge lengths by a random number between the following ranges: $[1, 3], [1, 5], [1, 10], [3, 5], [3, 10], [5, 10]$ without the scaling step (to set the diameter of the tree) afterwards. Therefore the maximum pairwise inversion distance is higher if the tree has more taxa. Second, the parameters for the GNT model is slightly different. Finally, we put the results using different numbers of taxa in the same figure; the number of taxa is either 40 (37 genes and 120 genes), 80 (120 genes only), or 160 (120 genes only). However the results do not affect the conclusion we draw here.

   In Figure 5.14, the left figures are the results using 37 genes, and the right figures are the results using 120 genes. Despite the higher numbers of taxa (which we showed to lower the accuracy previously), datasets with 120 genes tend to yield more accurate trees than datasets with 37 genes. If we look at the number of different gene orders in each setting:

$$\begin{aligned} 2^{37-1}(37-1)! &= 2.556 \times 10^{52} \\ 2^{120-1}(120-1)! &= 3.705 \times 10^{232} \end{aligned}$$

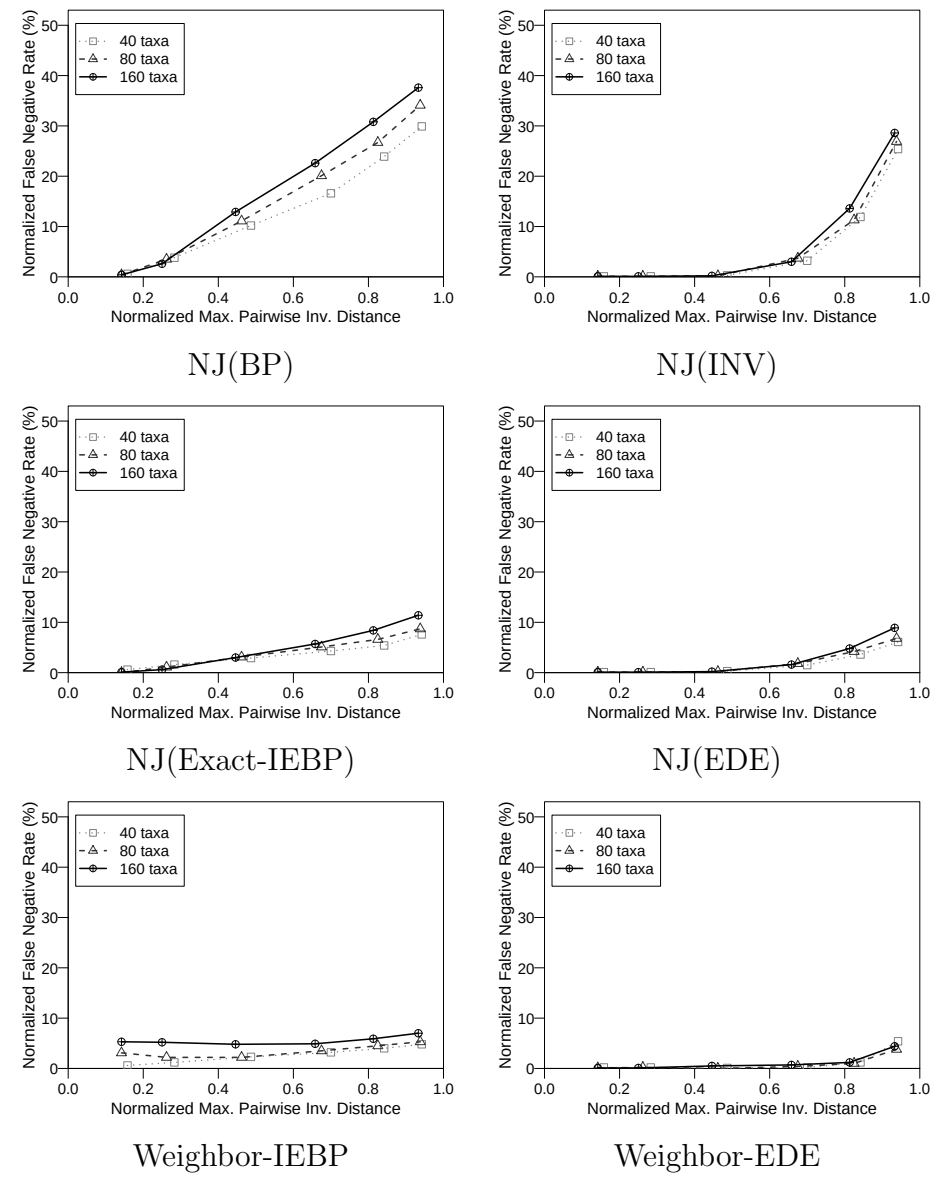


Figure 5.12: The accuracy of the reconstructed tree versus the number of taxa. The number of genes is 120, and the GNT model is inversion only: $(\alpha, \beta) = (0, 0)$.

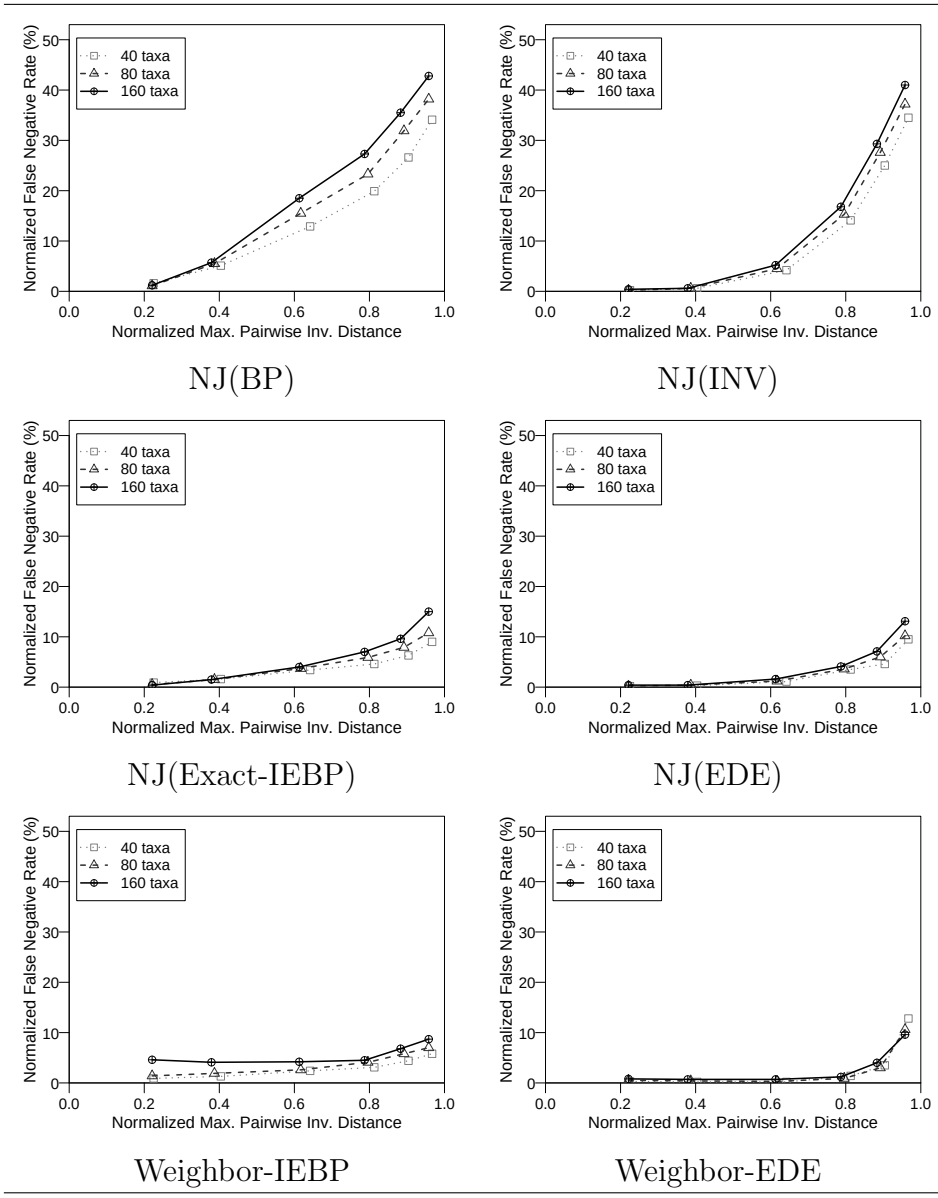


Figure 5.13: The accuracy of the reconstructed tree versus the number of taxa. The number of genes is 120, and the GNT model is transpositions and inverted transpositions only: $(\alpha, \beta) = (\frac{1}{2}, \frac{1}{2})$.

the number of states is much higher with 120 genes.

Running time NJ, BioNJ-IEBP, and BioNJ-EDE all finish within 1 second for all settings on our Pentium workstations running Linux. However, Weighbor-IEBP and Weighbor-EDE take considerably more time; both methods take about 10 minutes to finish when the number of genomes is 160.

5.4 The robustness of NJ(Exact-IEBP) and Weighbor-IEBP to parameter misspecification

Both NJ(Exact-IEBP) and Weighbor-IEBP require the parameters of the GNT model (i.e. the relative probabilities of the three types of events), but it is usually not known beforehand. In this section we demonstrate the robustness of the NJ(Exact-IEBP) and Weighbor-IEBP estimator when the model parameters are unknown. The robustness of the NJ(Approx-IEBP) is similar to that of NJ(Exact-IEBP). The settings are the same in Table 5.1. The experiment is similar to the previous experiment, except here we use both the correct and the incorrect values of (α, β) for the computation of Exact-IEBP distance, and its variance when Weighbor is used. The results are in Figures 5.15 and 5.16: the accuracies are similar whether we use the correct parameters or not, except for Weighbor-IEBP when the GNT model is inversion only, and the rate of evolution is very low. These results suggest both methods are robust against errors in (α, β).

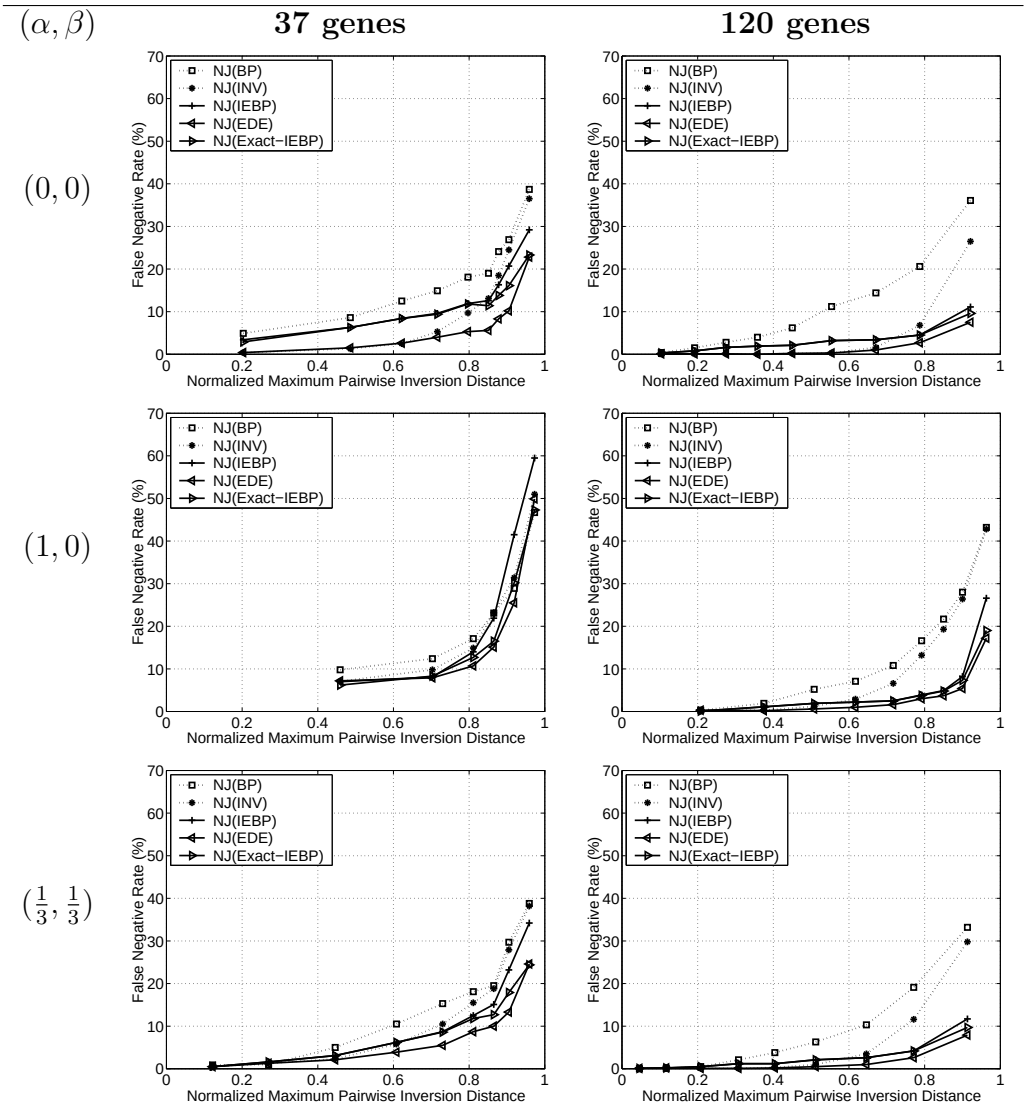


Figure 5.14: The accuracy of distance-based tree reconstruction versus the number of genes in each genome.

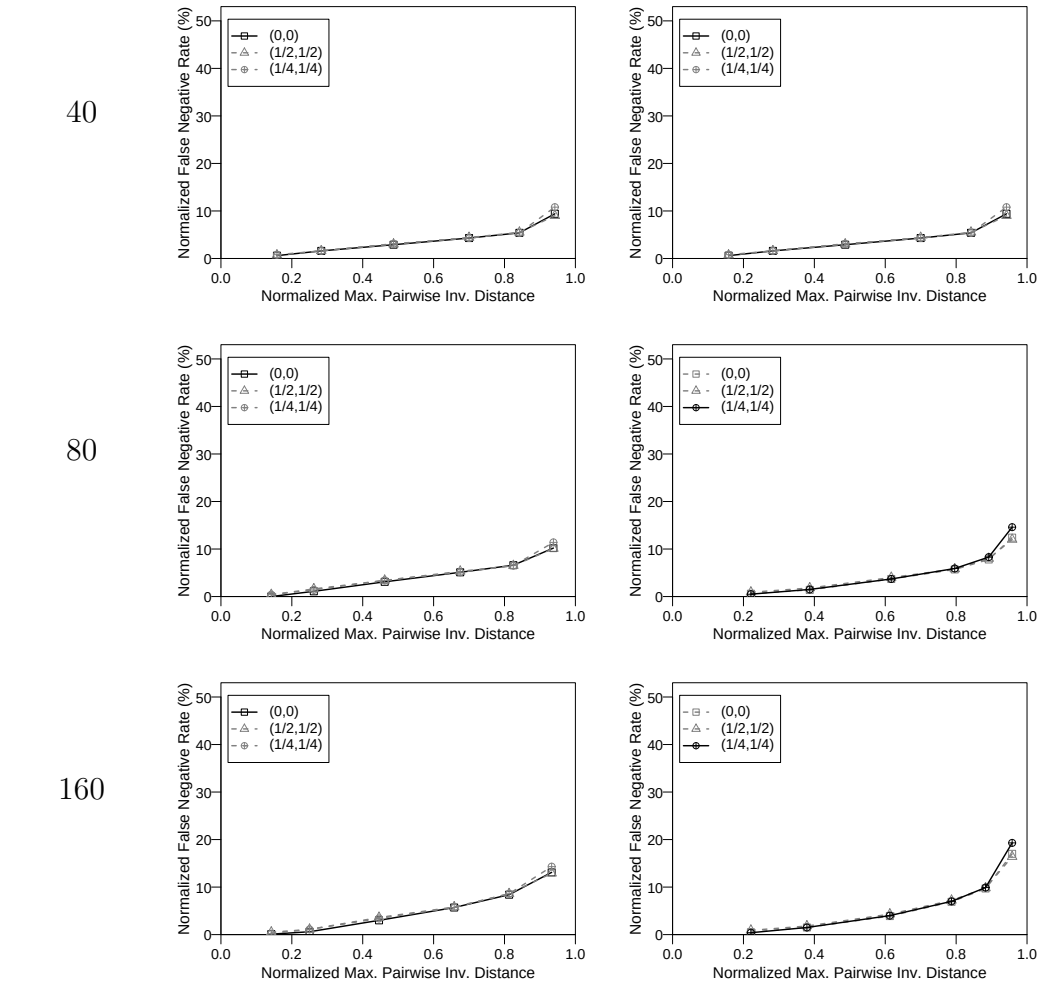


Figure 5.15: Robustness of the `Exact-IEBP` method to unknown parameters (see Section 5.4). The two values in the legend are the α and β values used in the `Exact-IEBP` method. The probability a rearrangement event is an inversion, a transposition, or an inverted transposition is $1 - \alpha - \beta$, α, and β, respectively.

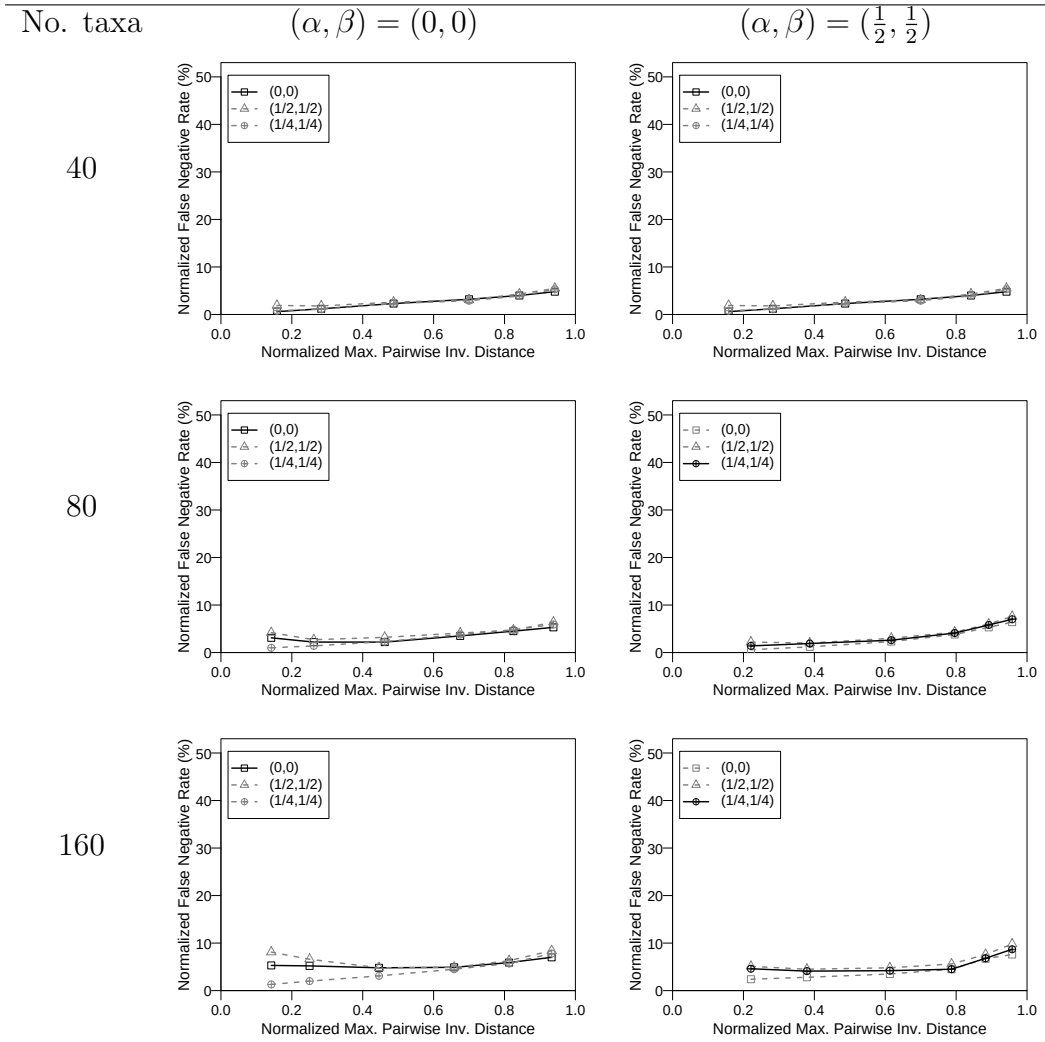


Figure 5.16: Robustness of the `Weighbor-IEBP` method to unknown parameters (see Section 5.4). The two values in the legend are the α and β values used in the `Exact-IEBP` method. The probability a rearrangement event is an inversion, a transposition, or an inverted transposition is $1 - \alpha - \beta$, α, and β, respectively.

Chapter 6

Genome Rearrangement Phylogeny Using Parsimony Criteria

6.1 Parsimony-based Methods using adjacency encodings

All methods discussed in this chapter are based on adjacency encodings generated from the signed permutation. These character matrices are then subjected to parsimony searches—for which good implementations have long been available.

The *Maximum Parsimony on Binary Encodings* (`MPBE-1`, also called `MPBE` in [20]) algorithm [20, 69] has exponential running time in the number of genomes (because the parsimony problem is NP-hard), but runs very fast in practice. In `MPBE-1`, each gene ordering is translated into a binary sequence, where each site from the binary sequence corresponds to a pair of genes. (The ordering of the sites is immaterial in this encoding.) For the pair (g_i, g_j), the sequence has a 1 at the corresponding site if g_i is immediately followed by g_j in the gene ordering and a 0 otherwise (note that g_i and g_j can be negative and that, since (g_i, g_j) and $(-g_j, -g_i)$ denote the same adjacency, we need only one site for both). There are $\binom{n}{2}$ pairs, where n is the number of genes in each genome, but we drop the sites where every sequence has the same value.

[1] The content of this chapter also appeared in [85].

The other encoding, MPBE-2, is a modification of MPBE-1: the characters in the MPBE-2 encoding is a subset of the set of adjacency pairs of an MPBE-1 encoding. The motivation is to reject those shared characters in the dataset because they are in the ancestral genome. The result is we restrict the datasets to shared derived characters, since they are the characters that are informative about the bipartitions of the phylogeny. In the MPBE-2 encoding, the gene order of the root is assumed. We then drop all characters that are adjacency gene pairs in the root gene order in the MPBE-1 strings.

Bryant [12] proposed an encoding method, based on an earlier characterization approach of Boore [10], that we have used to develop a new character scoring method that we call *Maximum Parsimony on Multistate Encodings* (MPME). Let n be the number of genes in each genome; then each gene order is translated into a sequence of length $2n$. For every i, $1 \leq i \leq n$, site i takes the value of the gene immediately following gene i and site $n + i$ takes the value of the gene immediately following gene -i. For example, the circular gene ordering (1,-4,-3,-2) corresponds to the MPME sequence of (-4, 3, 4,-1, 2, 1,-2,-3). Bryant showed in [12] that the MPME score of any binary tree T is a tighter lower bound of the breakpoint length of T than the MPBE-1 score of T.

The difficulty of MPME is the fact the number of states per character is not a constant, but linear in the number of genes. Each site can take up to $2(n - 1)$ different values; the unbounded number of states per characters is a drawback in practical implementations, which usually assume that this number is bounded by a small constant (for example, the bound is 32 in PAUP* 4.0 [78]). Even after remapping the set of successor values into a consecutive set of symbols, the number of symbols often exceeds the PAUP bound for larger problems. We could decompose each multistate character into a collection of

new characters with fewer states and thus avoid the limitation at the cost of longer running times—we will explore this strategy in future work.

Figure 6.1 contains examples of the three encodings.

Running time for scoring a tree According to Theorem 2 in Chapter 2, the running time for scoring a tree is bounded above asymptotically by the product of the number of states per character (site), the number of characters, and the number of taxa in the dataset. It is clear we can compute the encodings of `MPBE-1`, `MPBE-2` (given the assumption of the root gene order), and `MPME` in time $O(k^2n)$, where n is the number of taxa, and k is the number of genes in each genome. In `MPBE-1`, the number of characters is $O(k^2)$, the number of states per character is constant (either 0 or 1). Therefore, the running time for computing the `MPBE-1` score of a tree is $O(k^2n)$. In `MPBE-2`, the number of characters is $O(k)$ less than that of `MPBE-1`, so the running time bound is not changed. Finally, in `MPME` there are $2k$ characters. Each character can have up to $2(k-2)$ states. The running time for scoring the `MPME` tree is also $O(2(k-2) \cdot 2k \cdot n) = O(k^2n)$.

Theorem 12. *Let us be given n signed circular or linear genomes having the same set of k distinct genes, and any binary tree topology T whose leaves are the n genomes. We can compute the `MPBE-1`, `MPBE-2`, and the `MPME` scores in $O(k^2n)$ time.*

6.2 Design of the Experiments

The goal of our experiments is to compare the tradeoffs (time vs. accuracy) offered by NJ with those offered by the parsimony-based methods; thus we present results for both running time and accuracy.

Genome	Gene order (circular)						Reversed equivalent representation					
A	1	2	3	4	5	6	-6	-5	-4	-3	-2	-1
B	1	2	-5	-4	3	6	-6	-3	4	5	-2	-1
C	1	-6	-5	-4	-3	-2	2	3	4	5	6	-1

(a) Signed circular genomes

Genome	Adjacencies										
	(1,2)	(2,3)	(3,4)	(4,5)	(5,6)	(6,1)	(2,-5)	(-4,3)	(3,6)	(1,-6)	(-2,1)
A	1	1	1	1	1	1	0	0	0	0	0
B	1	0	0	1	0	1	1	1	1	0	0
C	0	1	1	1	1	0	0	0	0	1	1

(b) MPBE-1

Genome	Adjacencies										
	(1,2)	(2,3)	(3,4)	(4,5)	(5,6)	(6,1)	(2,-5)	(-4,3)	(3,6)	(1,-6)	(-2,1)
A							0	0	0	0	0
B							1	1	1	0	0
C							0	0	0	1	1

The first six adjacency pairs are in the ancestor (genome A) and are removed in the `MPBE-2` encoding. (c) MPBE-2

Genome	Signed genes											
	1	2	3	4	5	6	-1	-2	-3	-4	-5	-6
A	2	3	4	5	6	1	-6	-1	-2	-3	-4	-5
B	2	-5	6	5	-2	1	-6	-1	4	3	-4	-3
C	-6	3	4	5	6	-1	2	1	-2	-3	-4	-5

(d) MPME

Figure 6.1: Examples of the three encodings of genome rearrangements, `MPBE-1`, `MPBE-2`, and `MPME`. See Section 6.1 for details.

6.2.1 Quantifying Accuracy

Given an inferred tree, we assess its *topological accuracy* by computing the *normalized false negative (FN) rate* with respect to the *true tree*. The true tree may not be the model tree itself: the evolutionary process may cause no changes on some edges of the model tree, in which case we define the true tree to be the result of *contracting* those edges in the model tree. For every tree there is a natural association between every edge e and the bipartition on the leaf set induced by deleting e from the tree. Let T be the true tree and let T' be the inferred tree. An edge of T is *missing* in T' if T' does not contain an edge defining the same bipartition; such an edge is then called a *false negative* (FN).

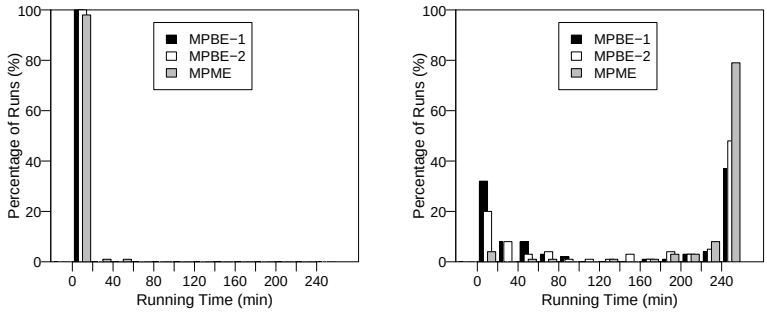
6.2.2 The Experiments

We use the same settings in Table 5.1 in Chapter 5. The procedure of the experiment follows that in Chapter 3. In addition, we compute the most parsimonious trees from the heuristic search using the three encodings (`MPBE-1`, `MPBE-2`, and `MPME`). When the parsimony search returns more than one tree, we use the majority-rule consensus for comparison to the true tree. We use PAUP* 4.0b8 [78] for NJ, to compute the false negative rate between two trees, and for the parsimony search using the three encodings. The setting of the parsimony heuristic search was as follows: the upper bound for the running time was 240 mins., the heuristic search uses Tree-Bisection-Reconnection (TBR) operations to generate new trees, at any time we held the 5 trees having the lowest parsimony score, and we use the NJ trees with our five distances as the starting trees. All experiments were conducted on the 16-processor Phylofarm cluster at The University of Texas at Austin.

6.3 Results of the Experiments

As mentioned, MPME will exceed 32 states per character for large problems. The problem worsens with increasing rate of evolution; for runs with 120 genes, 160 taxa, and edge length $[5, 10]$, PAUP *always* rejects the MPME data matrix. We ignore all MPME datasets rejected by PAUP; future experiments will investigate running these datasets with multistate characters replaced by sets of binary characters.

Figure 6.2 shows histograms of the running times of the parsimony-based methods for two sizes of problems; on smaller problems (40 taxa), the parsimony search ran quickly (20 mins.), but larger numbers of taxa caused sharp increases in running times—to the point where `MPME` generally reached the time limit. In comparison, the `NJ`-based methods ran faster—typically in 8 minutes or less, with no variation among runs using a particular estimator.



120 genes, 40 taxa 120 genes, 160 taxa

Figure 6.2: PAUP running times for the three parsimony-based methods. The vertical bars right of 240 minutes are the portions of the runs that exceed the parsimony search limit.

We present in Figures 6.3, 6.4, and 6.5 some of our results (the full set will be presented in the final dissertation). We show three different prob-

lem sizes, which we can think of as small, medium, and large. For 37 genes, both distance- and parsimony-based methods (except MPME) yield false negative rates of at least 10%—the low number of genes reduces the amount of phylogenetic information. For 120 genes, trees produced by parsimony-based methods and NJ using Approx-IEBP, Exact-IEBP, and EDE have false negative rates at most 20% (10% for higher rate and 40 taxa), and outperform NJ(INV) and NJ(BP) by a large margin when the amount of evolution is high.

While MPME usually produces the most accurate trees among the parsimony-based methods, it is considerably slower than MPBE-1; indeed, we expect its performance on larger datasets is time-limited—had we given it more time to run, it would have surpassed the other MP-based methods easily. With 37 genes, increasing the rate of evolution improves the accuracy of MPME, but worsens that of MPBE-1 and MPBE-2, whereas all three methods improve in accuracy for larger evolutionary rates with 120 genes.

NJ(EDE) is clearly the best distance-based method: not only is its accuracy equal or superior to that of others, it is also faster than all but the uncorrected methods. The three parsimony-based methods are as accurate as the best distance-based methods for low evolutionary rates and more accurate for high evolutionary rates, but also more expensive. MPME is the best among them: it behaves well at all rates and is much better at high rates in smaller data sets. Our results suggest that using an encoding that attempts to capture more details about the gene order (like MPME) preserves useful phylogenetic information that a parsimony-based search (with sufficient time) can put to good use.

6.4 Maximum Parsimony and Topological Accuracy

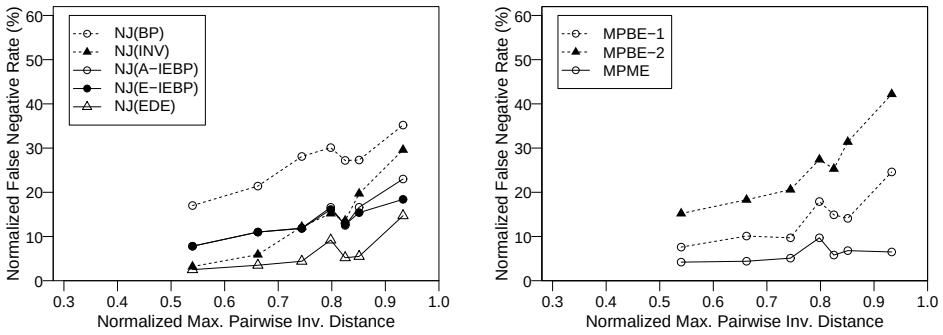
The main goal of phylogeny reconstruction is to produce the correct tree topology. Two basic approaches are currently used for phylogeny reconstruction from whole genomes: distance-based methods such as NJ applied to techniques for estimating distances and "maximum parsimony" (MP) approaches, which attempt to minimize the "length" of the tree, for a suitably defined measure of the length.

We examine two specific MP problems in this section: the breakpoint phylogeny problem, where we seek to minimize the total number of breakpoints over all tree edges, and the inversion phylogeny problem, where we seek to minimize the total number of inversions. We want to determine, using a simulation study, whether topological accuracy is improved by reducing the number of inversions or the number of breakpoints. If possible, we also want to determine whether the breakpoint phylogeny problem or the inversion phylogeny problem are topologically more accurate under certain evolutionary conditions, and if so, under which conditions.

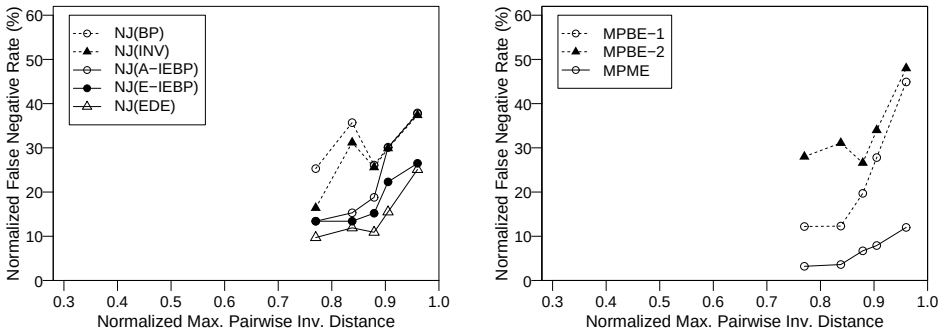
We ran a large series of tests on model trees to investigate the hypothesis that minimizing the total breakpoint distance or inversion length of trees would yield more topologically accurate trees. We ran NJ on a total of 209 datasets with both inversion and breakpoint distances. Each test consists of at least 12 data points, on sets of up to 40 genomes. We used two genome sizes (37 and 120 genes, representative of mitochondrial and chloroplast genomes, respectively) and various ratios of inversions to transpositions and inverted transpositions, as well as various rates of evolution. For each dataset, we computed the total inversion and breakpoint distances and compared their values with the percentage of errors (measured as false negatives).

We used the nonparametric *Cox-Stuart test* [19] for detecting trends—i.e., for testing whether reducing breakpoint or inversion distance consistently reduces topological errors. Using a 95% confidence level, we found that over 97% of the datasets with inversion distance and over 96% of those with breakpoint distance exhibited such a trend. Indeed, even at the 99.9% confidence level, over 82% of the datasets still exhibited such a trend.

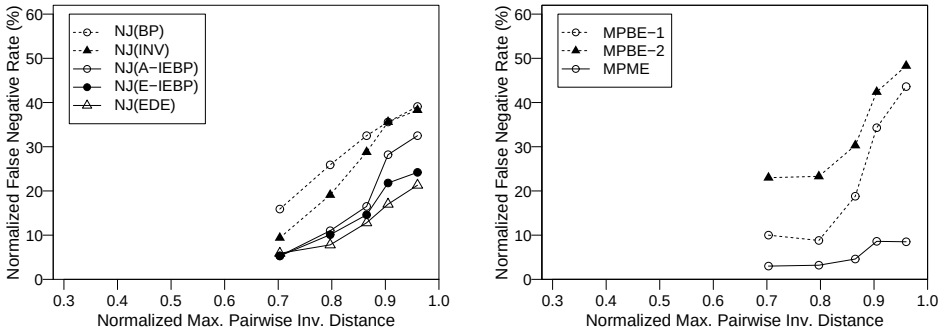
Figures 6.6 and 6.7 show the results of scoring the different NJ trees under the two optimization criteria: breakpoint score and inversion length of the tree. In general, the relative ordering and trend of the curves agree with their corresponding curves of Figure 5.4, suggesting that decreasing the number of inversions or breakpoints leads to an improvement in topological accuracy. The correlation is strongest for the 120-gene case; this may be because, for the same number of events but a larger number of genes, the rate of evolution effectively goes down and overlap of events becomes less likely. Finally, this trend still holds under the other evolutionary models (such as when only transpositions occur).



$(\alpha, \beta) = (0, 0)$

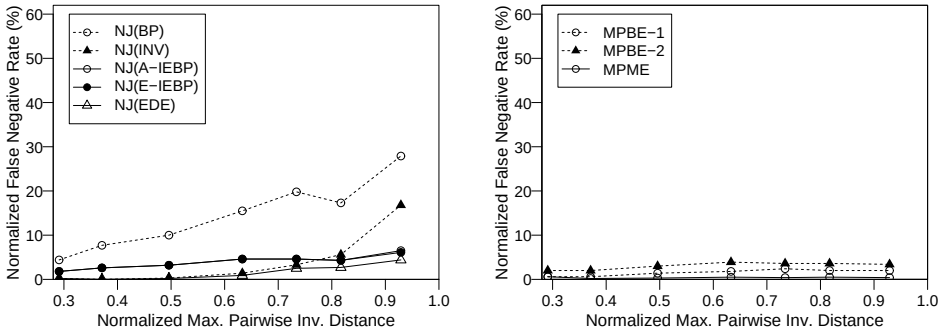


$(\alpha, \beta) = (\frac{1}{2}, \frac{1}{2})$

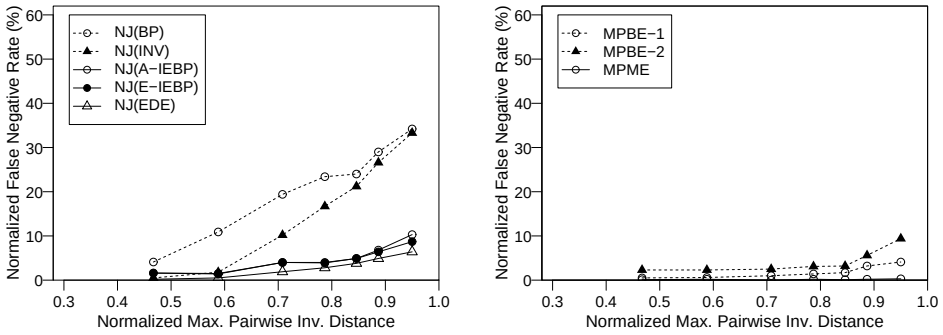


$(\alpha, \beta) = (\frac{1}{3}, \frac{1}{3})$

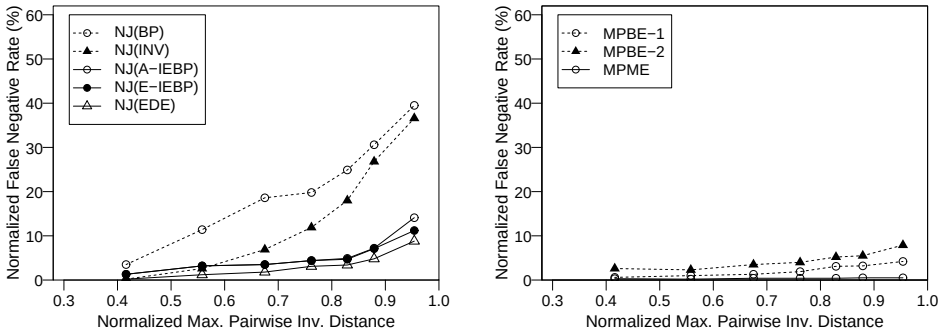
Figure 6.3: Topological accuracy of phylogenetic methods on problems with 37 genes and 40 taxa. The x-axis is normalized by the number of genes, the highest inversion distance two gene orders can have. Our plots result from binning the values into range of evolutionary distances (maximum pairwise inversion distance in the dataset) and plotting the average value for each bin.



$(\alpha, \beta) = (0, 0)$

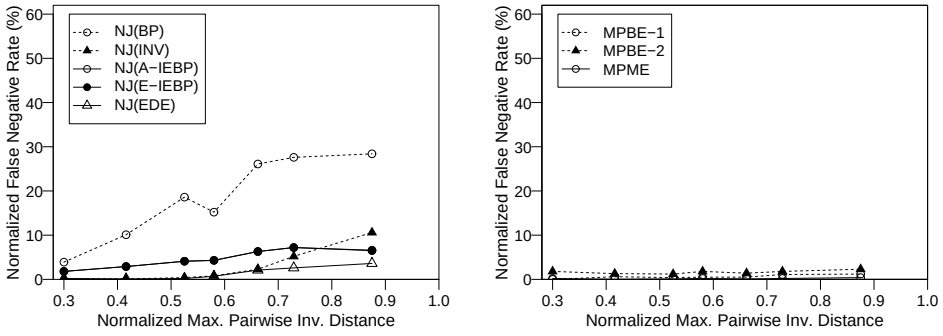


$(\alpha, \beta) = (\frac{1}{2}, \frac{1}{2})$

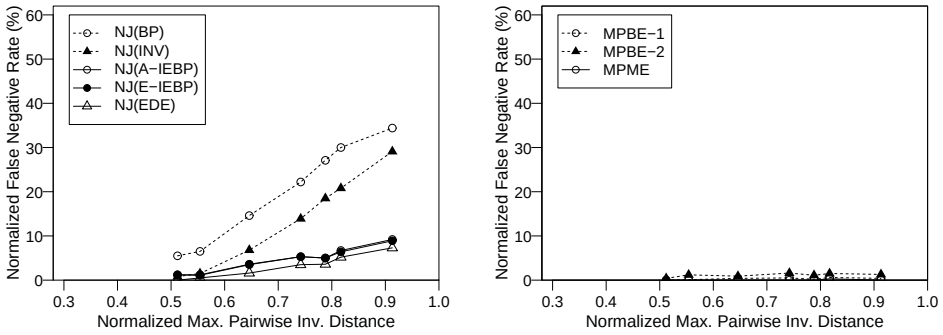


$(\alpha, \beta) = (\frac{1}{3}, \frac{1}{3})$

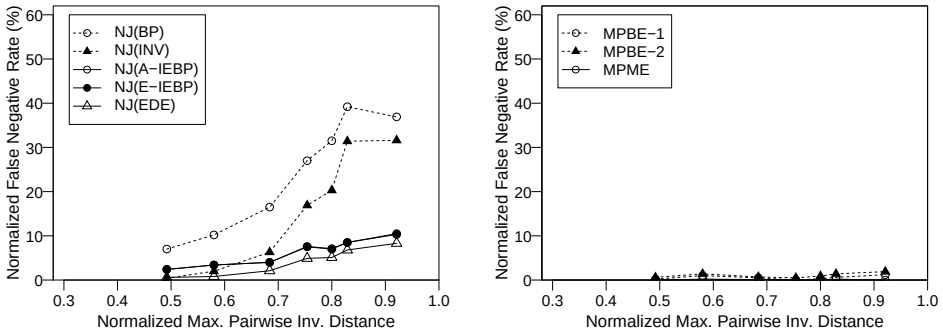
Figure 6.4: Topological accuracy of phylogenetic methods on problems with 120 genes and 40 taxa.



$(\alpha, \beta) = (0, 0)$

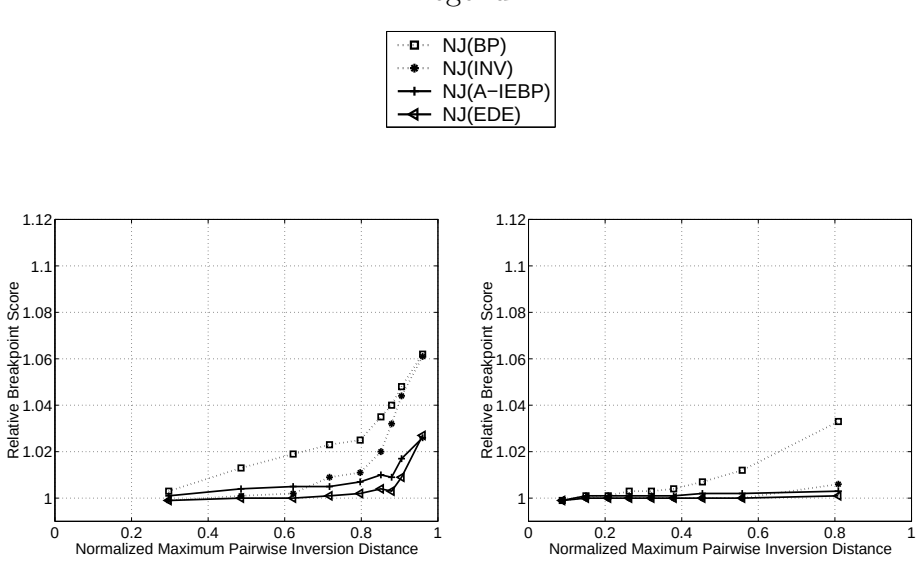


$(\alpha, \beta) = (\frac{1}{2}, \frac{1}{2})$



$(\alpha, \beta) = (\frac{1}{3}, \frac{1}{3})$

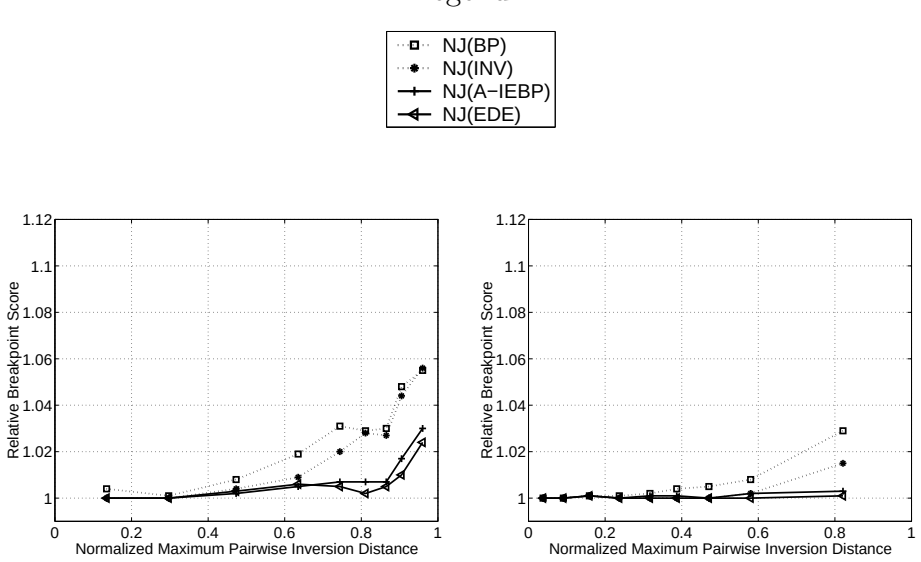
Figure 6.5: Topological accuracy of phylogenetic methods on problems with 120 genes and 160 taxa.



(a) breakpoint score, 37 genes (b) breakpoint score, 120 genes

(c) inversion length, 37 genes (d) inversion length, 120 genes

Figure 6.6: Scoring NJ methods under various distance estimators as a function of the maximum pairwise inversion distance for 10, 20, and 40 genomes. Plotted is the ratio of the NJ tree score to the model tree score (breakpoint or inversion) on an inversion-only model tree.



(a) breakpoint score, 37 genes (b) breakpoint score, 120 genes

(c) inversion length, 37 genes (d) inversion length, 120 genes

Figure 6.7: Scoring NJ methods under various distance estimators as a function of the maximum pairwise inversion distance for 10, 20, and 40 genomes. Plotted is the ratio of the NJ tree score to the model tree score (breakpoint or inversion) on a model tree where the three classes of events are equiprobable.

Chapter 7

Statistically Based Postprocessing of Phylogenetic Analysis by Clustering

7.1 Introduction

Many tree reconstruction methods produce more than one candidate tree for the input dataset. For example, the *maximum parsimony* [79] method returns those binary trees with the lowest parsimony score. Very often the number of trees can be in the hundreds or thousands. If this is the case, a consensus tree of the candidate trees is computed so as to resolve the conflict, summarize the information, and reduce the overwhelming number of possible solutions to the evolutionary history.

Many consensus tree methods are available; among them the strict consensus, majority consensus, and Adams consensus are the most popular [1, 45]. A common feature of these methods is they all produce one tree. There are several shortcomings of this approach. First, a single tree loses a lot of information about the set of candidate trees, including how the trees are distributed in the space of all binary trees, and how the trees are similar to each other. In addition, returning only one tree can make postprocessing very sensitive to the input. Several studies show the single-consensus method is limited [48, 74]. The two cited papers all point out that when given certain reasonable and de-

[1]The content of this chapter also appeared in [77].

sirable conditions, no single-tree consensus methods satisfy them at the same time (the phylogenetic version of Arrow's Impossibility Theorem [3]).

In this chapter we present a different approach to postprocessing. The set of candidate trees is divided into several subsets using clustering methods. Each cluster is then characterized by its own consensus tree. We pose several theoretical optimization problems for these kinds of outputs, and present some initial progress on these problems; these are presented in Section 7.3. The rest of this chapter is focused on an empirical study; we present our results in Section 7.4.

7.2 Notation

Let $S = \{1, 2, \ldots, n\}$ denote the n taxa being studied. Let $\mathcal{T}_n$ denote the set of all (unrooted) binary trees with S as their leaf set. The cardinality (number of elements) of $\mathcal{T}_n$ is $|\mathcal{T}_n| = (2n-3)!! = 1 \cdot 3 \cdot 5 \cdots (2n-3)$.

Let T denote the set of input trees (e.g. the most parsimonious trees from a maximum parsimony analysis). We make the following definitions and use the following notations:

1. A **clustering** $\mathbf{C}$ of T is a partition of T. A clustering $\mathbf{C}$ is *covering* if for every tree $t \in T$, t is in at least one cluster $C \in \mathbf{C}$; if not, $\mathbf{C}$ is *noncovering*. Each member C of $\mathbf{C}$ is a cluster.

2. Let $SC(C)$ denote the strict consensus of all trees in C (recall the strict consensus of a set of trees T is the tree whose edges are in every tree of T).

3. The **bounding ball** $B(C)$ of a cluster C is defined by $B(C) = \{t \in \mathfrak{T}_n : t \geq SC(C)\}$, i.e. the set of all binary refinements of the strict consensus of C; we let $B(\mathbf{C}) = \cup_{C \in \mathbf{C}} B(C)$.

4. Given a set of trees T, and two clusterings of T, $\mathbf{C}$ and $\mathbf{C}'$, we say $\mathbf{C}$ is a **refinement** of $\mathbf{C}'$ (or $\mathbf{C}$ *refines* $\mathbf{C}'$) if every member of $\mathbf{C}'$ is a union of member(s) in $\mathbf{C}$.

5. We use $d(t, t') = d_{RF}(t, t')$ to denote the Robinson-Foulds distance between two trees t and t'. See Section 2.2.1 for the definition of the Robinson-Foulds distance.

7.3 Criteria for Clustering in the Tree Space

In this section we describe the criteria used for clustering phylogenetic trees.

7.3.1 Biologically based criteria

Parameters for clustering Given a cluster C, we define the following parameters of C:

1. $\text{diam}(C) = \max_{t,t' \in C} d(t, t')$ is the **diameter** of C.

2. $\lambda(C) = \frac{|E(SC(C))|}{n-3}$ is the **specificity** of C; it is the normalized number of internal edges of the strict consensus of C.

3. $\rho(C) = \frac{|C|}{|B(C)|}$ is the **density** of C.

Biologists are interested in the specificity; the higher it is, the more information is present. This value is related to the diameter since it is easy to show the

following:

Lemma 4.

$$1 - \frac{diam(C)}{2(n-3)} \geq \lambda(C).$$

Proof. Let a be the number of internal edges of $SC(C)$. Pick any pair of (binary) trees t and t' in C. The strict consensus of t and t' must refine $SC(C)$. But the number of internal edges of their strict consensus is $(n-3) - \frac{d(t,t')}{2}$. Therefore

$$(n-3)\lambda(C) = a \leq (n-3) - \frac{d(t,t')}{2}$$

Divide both sides by $(n-3)$ and we have the lemma. □

The density reflects the support of the cluster's evolutionary hypothesis, which is represented by the consensus tree.

Based on these parameters we can define the parameters of the whole clustering $\mathbf{C}$. Let $f(C)$ be a parameter value of cluster C. We have

1. $M(\mathbf{C}; f) = \max_{C \in \mathbf{C}} f(C)$ (the maximum value of f over all clusters).
2. $m(\mathbf{C}; f) = \min_{C \in \mathbf{C}} f(C)$ (the minimum value of f over all clusters).
3. $W(\mathbf{C}; f) = \frac{1}{|\mathbf{C}|} \sum_{C \in \mathbf{C}} |C| f(C)$ (the weighted value of f over all clusters).

The number of clusters is also an important parameter.

Some of the parameters are good criteria for a clustering problem; some are not. For example, the maximum density and the minimum diameter can be optimized easily by picking one tree arbitrarily and making it its own cluster. Some parameters can be ambiguous. For example, though we favor clusters with small diameters since the specificity is higher, a dense cluster with large

diameter means the number of trees in the cluster is very large. Whether we favor a cluster with large diameter over a cluster with small diameter really depends on how the input set of trees relate to each other.

Bicriterion problems Even though some parameters pass the above test, we can still have trivial solutions. The two obvious trivial solutions are the *single-cluster clustering* (placing all trees in one cluster) and the *single-tree clustering* (placing each tree in its own cluster). The first clustering automatically maximizes the minimum diameter, while the second maximizes the minimum density and specificity. To avoid such solutions we look at *bicriterion problems*, problems that consider two parameters at once. For example, we can minimize $M(\mathbf{C}; \text{diam}) + aW(\mathbf{C}; \lambda)$, a linear combination of the maximum diameter and the weighted sum of specificities.

Bicriterion problems involving k, the number of clusters are most natural and interesting. Since k is bounded by the number of trees in the input, we can find the solution that optimizes the other parameter in the problem for each value of k, and choose the clustering that best optimizes the objective function. Also note when we refine a clustering by dividing some of the clusters, the diameter of each new cluster is smaller or equal to the diameter of the original cluster – the minimum, maximum, and weighted sum of diameters go down. Similarly the minimum, maximum, and weighted sum of specificity go up.

Observation 1. *The minimum, maximum, and weighted sum of diameters or specificity are* monotone *with respect to refinements of clusterings.*

Therefore by dividing clusters, the clustering generally has better performance. As we will see in Section 7.4 the score for each clustering obtained

by agglomerative clustering improves as the number of clusters increases, but this is not true for every method.

7.3.2 Statistically based criteria

Biologists assume the true tree is among the tree obtained during phylogenetic analysis; without any additional information, all trees are considered equally likely to be the true tree. Thus, the set of trees defines a probability distribution on tree space. Because the number of trees can be overwhelming, biologists replace them with their strict consensus tree, and the original output trees are then ignored. Knowing only that the true tree refines this consensus tree, then, we have another probability distribution, with every binary tree that refines the consensus tree considered equally likely to be the true tree.

Our objective, then, is to increase the number of consensus trees to a still tolerably small number so that the probability distribution defined by these trees is closer to that of the original output. We will look at the following bicriterion problem, called *complexity vs. information content*. The complexity of a clustering can be measured in several ways, such as the total number of edges in the strict consensus trees of each cluster, but we will use the number of clusters.

We now introduce two criteria that capture this concept. The first, called *information loss*, is introduced by the author that is specially designed for the tree space. The second, called *information bottleneck*, is first introduced in [80]; it is designed for general purpose clustering problems and does not take into consideration the structure of the tree space.

7.3.3 Information loss

To evaluate the *information* conveyed in a clustering $\mathbf{C}$, we define a distribution on the tree space $\mathcal{T}_n$ for $\mathbf{C}$. Consider the original set T of m binary trees, each of them having the same probability of being the true tree. The corresponding distribution is

$$f(t) = \begin{cases} \frac{1}{m} & \text{if } t \in T \\ 0 & \text{if } t \notin T \end{cases}$$

Let $\mathbf{C}$ be a particular clustering of T. Let $c_i = |C_i|$. Note the clusters may have overlapping bounding balls. Let $B = \cup_{C \in \mathbf{C}} B(C)$ be the union of bounding balls, and let $b = |B|$. If we assume we cannot distinguish between these trees, we can define a distribution as follows:

$$g(t) = \begin{cases} \frac{1}{b} & \text{if } t \in B \\ 0 & \text{if } t \notin B \end{cases}$$

We call this the *uniform* distribution. Note distributions f and g agree if $\mathbf{C}$ is such that every tree in T is in its own cluster, meaning there is no information loss in $\mathbf{C}$.

Information loss We define the *information loss* as the distance between the distributions of two clusterings. Let f and g be the distributions of the original set of trees and the clustering of the input, respectively. The most popular distances are

1. L_∞ distance: $L_\infty(f, g) = \max_{t \in \mathcal{T}_n} |f(t) - g(t)|$.
2. L_1 distance: $L_1(f, g) = \sum_{t \in \mathcal{T}_n} |f(t) - g(t)|$.

3. L_2 distance: $L_2(f,g) = \sqrt{\sum_{t \in \mathcal{T}_n} (f(t) - g(t))^2}$.

Note that if $T = B(\mathbf{C})$ then the distances are 0 for the uniform distribution.

Another popular distance between distributions is the *Kullback-Leibler* (KL) distance [42].

$$H(g|f) = \sum_{t \in \mathcal{T}_n} f(t) \ln \frac{f(t)}{g(t)}$$

Since the set of trees having nonzero value for f (the support of f) is T, we have

$$H(g|f) = \sum_{t \in T} f(t) \ln \frac{f(t)}{g(t)}$$

The KL distance is not symmetric. The technical difficulty of the KL distance approach is that there may be trees t such that $f(t) \neq 0$ but $g(t) = 0$, so the ratio $\frac{f(t)}{g(t)}$ is not finite. We avoid this difficulty by assuming T is covered by $\mathbf{C}$ ($T \subseteq B(\mathbf{C})$). On the other hand, for trees $t \in B(\mathbf{C}) - T$, $f(t) = 0$ and $g(t) \neq 0$, we set $f(t) \ln \frac{f(t)}{g(t)} = 0$ (this is also based on the observation $\lim_{n \to \infty} \frac{1}{n} \ln \frac{1}{n} = 0$).

We note

$$\begin{aligned} H(g|f) &= \sum_{t \in \mathcal{T}_n} f(t) \ln \frac{f(t)}{g(t)} \\ &= \sum_{t \in T \cap B} \frac{1}{m} \ln \frac{1/m}{1/b} + \sum_{t \in B - T} 0 \\ &= \frac{m}{m} \ln \frac{b}{m} = \ln \frac{b}{m} \end{aligned}$$

Since we assume $b \geq m$, the distance is minimized (0) when $b = m$.

Theorem 13. *Among all clusterings* $\mathbf{C}$ *satisfying* $T \subseteq B(\mathbf{C})$*, a clustering* $\mathbf{C}^*$ *that minimizes* $|B(\mathbf{C})|$ *has minimal KL distance.*

Corollary 4. *The Kullback-Leibler distance is monotone with respect to refinements of clusterings.*

In [88] the information content of a single consensus tree is discussed.

7.3.4 Representative tree

In this section we look at the *representative set* problem: we want to find a small set of trees as representatives of the original set of trees, so the induced distribution is closest to the original distribution. We define the problem of finding one representative tree formally and show it can be solved in polynomial time using uniform distribution and the distances we defined in the previous section.

The single-representative tree problem is as follows. Assume we intend to use tree t to replace the whole set of trees T. Let $B(t)$ be the set of binary trees that refine t. The uniform distribution introduced by t is such that all trees in $B(t)$ has the same probability, and all trees outside t has zero probability. The information loss of t is defined similarly as that of clustering.

First let us assume all trees in T must be covered, i.e. $T \subseteq B$. Then $b = |B| \geq |T| = m$. From the discussion on the KL distance above, we see the strict consensus tree minimizes the KL distance. One can also show the strict consensus tree minimizes the KL distance as well as the L_1 and L_2 distances, since the strict consensus is the representative tree that covers T and minimizes $|B|$:

$$
\begin{aligned}
L_1(f,g) &= \sum_{t\in B} |f(t)-g(t)| \\
&= \sum_{t\in T} |\frac{1}{m}-\frac{1}{b}| + \sum_{t\in B-T} |0-\frac{1}{b}| \\
&= m(\frac{1}{m}-\frac{1}{b}) + (b-m)\frac{1}{b} \\
&= 2(1-\frac{m}{b}) \\
L_2(f,g)^2 &= \sum_{t\in B} (f(t)-g(t))^2 \\
&= \sum_{t\in T} (\frac{1}{m}-\frac{1}{b})^2 + \sum_{t\in B-T} (0-\frac{1}{b})^2 \\
&= m(\frac{1}{m^2}+\frac{1}{b^2}-\frac{2}{mb}) + (b-m)\frac{1}{b^2} \\
&= \frac{1}{m}+\frac{m}{b^2}-\frac{2}{b}+\frac{1}{b}-\frac{m}{b^2} \\
&= \frac{1}{m}-\frac{1}{b}
\end{aligned}
$$

Under the L_∞ distance the story is slightly different:

$$
\begin{aligned}
L_\infty(f,g) &= \max_{t\in\mathcal{T}_n} |f(t)-g(t)| \\
&= \max\{1(T-B)|\frac{1}{m}|, 1(B-T)|\frac{1}{b}|, 1(C)|\frac{1}{m}-\frac{1}{b}|\} \\
&= \max\{\frac{1}{m}1(T-B), \frac{1}{b}1(B-T), 1(C)|\frac{1}{m}-\frac{1}{b}|\}
\end{aligned}
$$

Here the function $1(X)$ is defined as follows: $1(X)=1$ if $X\neq\phi$, and $1(X)=0$ if $X=\phi$ (ϕ is the empty set.) If $B=T$, then $L_\infty(f,g)=0$. If $T\subseteq B$ (T is covered) then $b\geq m$ and $L_\infty(f,g)=\max\{\frac{1}{b}1(B-T), 1(C)|\frac{1}{m}-\frac{1}{b}|\}\leq\frac{1}{m}$.

If $T\subset B$ ($T\neq B$), the L_∞ distance is minimized if $b=2m$. A simple algorithm that finds the optimal representative tree under the L_∞ distance is

as follows. First compute the strict consensus, $SC(T)$, and the corresponding density. If the density is below 0.5, the problem is solved; otherwise, find an edge of the $SC(T)$ such that when contracted from $SC(T)$ the new tree has the smallest number of refinements (the ratio of increase of the number of refinements when an edge (u, v) is contracted can be determined solely by the degrees of u and v.) Let n be the number of leaves in each tree in T. Computing the strict consensus of T takes $O(nm)$ time [72] and finding the edge with minimal increase in the number of refinements takes $O(n)$ time. We have the following theorem:

Theorem 14. *Let T be a set of binary trees with the same set of leaves $\{1, 2, \ldots, n\}$. We use the uniform distribution in measuring the information loss, and require the representative tree to cover all trees in the input set of trees T.*

1. *The strict consensus of T is the representative tree of T with respect to L_1, L_2, and KL distance.*

2. *The representative tree of T with respect to L_∞ distance can be computed in $O(n|T|)$ time.*

3. *The strict consensus tree is optimal with respect to L_∞ distance even when we allow noncovering representative trees.*

Note we allow the case when there exist tree(s) from T that do not refine the representative tree, i.e. $T - B \neq \phi$. For the L_∞ distance, the difference of $f(t)$ and $g(t)$ for any $t \in T - B$ is $|f(t) - g(t)| = |\frac{1}{m} - 0| = \frac{1}{m}$. Since in the preceding paragraph we show that if $T \subseteq B$ then $L_\infty(f, g) \leq \frac{1}{m}$, the L_∞ distance is suboptimal if T is not covered.

Similarly one can prove the strict consensus optimizes the L_1 and L_2 distances if every tree in T is in the cluster and we allow only one cluster. Let C be the cluster and B be the bounding ball of C.

Recall that 1() is the indicator function. We extend the definition to handle set arguments: the function returns 1 if the argument is nonempty, 0 otherwise.

L_∞ distance The L_∞ distance is

$$\begin{aligned} & L_\infty(f, g_1) = \max_{t \in \mathcal{T}_n} |f(t) - g(t)| \\ = \; & \max\{1(T-B)|\frac{1}{m} - 0|, 1(B-T)|0 - \frac{1}{b}|, 1(C)|\frac{1}{m} - \frac{1}{b}|\} \\ = \; & \max\{\frac{1}{m} \times 1(T-B), \frac{1}{b} \times 1(B-T), 1(C) \times |\frac{1}{m} - \frac{1}{b}|\} \end{aligned}$$

If $B = T$, then $L_\infty(f, g) = 0$. If $T \subseteq B$ (T is covered) then $b \geq m$ and $L_\infty(f, g_1) = \max\{\frac{1}{b} \times 1(B-T), 1(C)| \times \frac{1}{m} - \frac{1}{b}|\} \leq \frac{1}{m}$. Thus a noncovering clustering ($T - B \neq \phi$) does not optimize the L_∞ distance with respect to the distribution of uniformity.

Assume $T \subseteq B$. The L_∞ distance is minimized if $b = 2m$.

L_1 distance Let $C = T \cap B = \cup_{i=1}^{k} C_i$, and let $c = |C|$. The L_1 distance is

$$\begin{aligned} L_1(f, g_1) &= \sum_{t \in \mathcal{T}_n} |f(t) - g(t)| \\ &= |T - B||\frac{1}{m} - 0| + |B - T||0 - \frac{1}{b}| + |C||\frac{1}{m} - \frac{1}{b}| \\ &= \frac{m-c}{m} + \frac{b-c}{b} + c|\frac{1}{m} - \frac{1}{b}| \end{aligned}$$

$$
\begin{aligned}
&= 1 - \frac{c}{m} + 1 - \frac{c}{b} + |\frac{c}{m} - \frac{c}{b}| \\
&= 2(1 - \frac{c}{\max\{m, b\}})
\end{aligned}
$$

L_2 distance The L_2 distance (squared) is

$$
\begin{aligned}
L_2(f, g_1)^2 &= \sum_{t \in \mathcal{T}_n} (f(t) - g(t))^2 \\
&= |T - B|(\frac{1}{m} - 0)^2 + |B - T|(0 - \frac{1}{b})^2 + |C|(\frac{1}{m} - \frac{1}{b})^2 \\
&= (m - c)\frac{1}{m^2} + (b - c)\frac{1}{b^2} + c(\frac{1}{m} - \frac{1}{b})^2 \\
&= \frac{1}{m} - \frac{c}{m^2} + \frac{1}{b} - \frac{c}{b^2} + \frac{c}{m^2} + \frac{c}{b^2} - \frac{2c}{mb} \\
&= \frac{1}{m} + \frac{1}{b} - \frac{2c}{mb} = \frac{m + b - 2c}{mb}
\end{aligned}
$$

A geometric interpretation is as follows. Let $\mathbf{m}$ be the vector having $|\mathcal{T}_n|$ components, where each component corresponds to a binary tree; we set $\mathbf{m}_t$ to be 1 if $t \in T$, 0 if $t \notin T$. We define the vector $\mathbf{b}$ similarly, except T is replaced by B. Then $L_2(f, g_1)^2 = \frac{||\mathbf{m}-\mathbf{b}||_2^2}{||\mathbf{m}||_2^2 ||\mathbf{b}||_2^2}$.

7.3.5 Information bottleneck

We now briefly describe the definition of the information bottleneck, and show how to compute the quantity for tree clustering.

Let X and Y be two random variables whose values are taken from two finite sets $\mathcal{X}$ and $\mathcal{Y}$, respectively. The mutual information is

$$
I(X; Y) = \sum_x \sum_y \Pr(X = x, Y = y) \log \frac{\Pr(X = x, Y = y)}{\Pr(X = x) \Pr(Y = y)}
$$

This quantity is symmetric and nonnegative; furthermore, $I(X, Y) = 0$ if and only if X and Y are independent.

Definition of information bottleneck We first use clustering of documents as an illustration. Let $\{d_1, \ldots, d_m\}$ be m distinct documents, and $\{w_1, \ldots, w_n\}$ be n distinct words. Let D and W be two random variables whose domains are the collection of documents and words, respectively; we let the joint distribution of D and W be $\Pr(D = d, W = w)$, which is computable from the input.

Let $\{c_1, \ldots, c_k\}$ be a k-partition over $\{d_1, \ldots, d_m\}$ (the c_i's are clusters). Let C be the random variable whose domain is the set of clusters. The joint distribution of C and W is

$$\Pr(C = c, W = w) = \sum_{d \in c} \Pr(D = d, W = w)$$

It can be shown that the mutual information between C and W is always smaller than the mutual information between D and W. The optimization criterion can be defined as follows: find a k-clustering over $\{d_1, \ldots, d_m\}$ so that the decrease in mutual information

$$I(D; W) - I(C; W)$$

is minimal.

Notations Again, let $T = \{t_1, t_2, \ldots, t_m\}$ be the m input binary trees, whose set of taxa is $\{1, 2, \ldots, n\}$. Let $E = \{e_1, \ldots, e_s\}$ be the s distinct nontrivial bipartitions from all trees in T. For every tree t, let $E(t)$ be the set of nontrivial bipartitions of t. Let $\nu(T, e)$ be the number of trees in T that have the bipartition e. We have $\sum_{e \in E} \nu(T, e) = m(n - 3)$.

Now let X_T be the random variable that takes its value from T, and Y_E be the random variable that takes its value from E, such that

$$\Pr(X_T = t, Y_E = e) = \begin{cases} \dfrac{1}{m(n-3)} & \text{if } t \in T \text{ and } e \in E(t) \\ 0 & \text{otherwise} \end{cases}$$

The uniform distribution is based on the following assumption: (1) all trees are binary (so all trees have the same number of bipartitions) and are treated equal, and (2) all bipartitions in each tree are treated equal.

From this we have $\Pr(X_T = t) = \frac{1}{m}$ if $t \in T$, $\Pr(X_T = t) = 0$ if $t \notin T$, and $\Pr(Y_E = e) = \frac{\nu(T,e)}{m(n-3)}$. The mutual information of X_T and Y_E is

$$\begin{aligned} I(X_T; Y_E) &= \sum_{t \in T} \sum_{e \in E(t)} \frac{1}{m(n-3)} \log \frac{\frac{1}{m(n-3)}}{\frac{1}{m}\frac{\nu(T,e)}{m(n-3)}} \\ &= \frac{1}{m(n-3)} \sum_{t \in T} \sum_{e \in E(t)} \log \frac{m}{\nu(T,e)} \\ &= \frac{1}{m(n-3)} \sum_{t \in T} \sum_{e \in E(t)} \log m - \frac{1}{m(n-3)} \sum_{t \in T} \sum_{e \in E(t)} \log \nu(T,e) \\ &= \log m - \frac{1}{m(n-3)} \sum_{t \in T} \sum_{e \in E(t)} \log \nu(T,e) \end{aligned}$$

Now consider the matrix M where the rows and columns are indexed by elements in T and E, such that:

$$M_{te} = \begin{cases} 1 & e \in E(t) \\ 0 & e \notin E(t) \end{cases}$$

Then $\nu(e)$ is the number of 1's in the column $M^{(e)}$.

Decrease in mutual information for tree clusterings Assume $\mathcal{C} = \{C_1, C_2, \ldots, C_k\}$ is a *hard* clustering over T, i.e. $\forall i, j,\ 1 \leq i < j \leq k$,

$C_i \cap C_j = \emptyset$. Let Z_C be a random variable taking its value from $\mathcal{C}$, such that $Z_C = C$ if $X_T = t$ and $t \in C$. Since $\mathcal{C}$ is a partition, we have

$$\Pr(Z_C = C) = \sum_{t \in C} \Pr(X_T = t) = \frac{|C|}{m}$$

Let $\nu(C, e)$ be the number of trees in C having bipartition e. The joint distribution of Z_C and Y_E is

$$\begin{aligned} \Pr(Z_C = C, Y_E = e) &= \sum_{t \in C} \Pr(X_T = t, Y_E = e) \\ &= \sum_{t \in C} \frac{1}{m(n-3)} = \frac{\nu(C, e)}{m(n-3)} \end{aligned}$$

To verify it indeed is a distribution:

$$\begin{aligned} \sum_C \sum_e \Pr(Z_C = C, Y_E = e) &= \sum_C \sum_e \frac{\nu(C, e)}{m(n-3)} \\ &= \sum_e \sum_C \frac{\nu(C, e)}{m(n-3)} = \sum_e \frac{\nu(T, e)}{m(n-3)} = 1 \end{aligned}$$

The mutual information of Y_E and Z_C is

$$\begin{aligned} I(Z_C; Y_E) &= \sum_C \sum_e \frac{\nu(C, e)}{m(n-3)} \log \frac{\frac{\nu(C,e)}{m(n-3)}}{\frac{|C|}{m} \frac{\nu(T,e)}{m(n-3)}} \\ &= \sum_C \sum_e \frac{\nu(C, e)}{m(n-3)} \log \frac{m\nu(C, e)}{|C|\nu(T, e)} \\ &= \sum_C \sum_e \frac{\nu(C, e)}{m(n-3)} \log \left(\frac{\nu(C, e)}{\nu(T, e)} / \frac{|C|}{|T|} \right) \end{aligned}$$

Return to the 0-1 matrix M. We create matrix N as follows: (1) the rows and the columns are indexed by elements of $\mathcal{C}$ and E, respectively, (2) the entries of N are

$$N_{Ce} = \sum_{t \in C} M_{te}$$

Then $M_{Ce} = \nu(C, e)$ and $\nu(T, e) = \sum_C \nu(C, e)$.

The decrease of mutual information due to clustering is

$$I(X_T; Y_E) - I(Z_C; Y_E)$$

Effect of trivial bipartitions Does including all-1 columns affect the decrease in mutual information? Clearly the ordering of the decrease in information does not change, i.e. a clustering with less decrease in mutual info also has a smaller decrease when we include all-1 columns. What about the change in magnitude?

Let us add v all-1 columns to the 0-1 matrix, and let the set $E'(t)$ denote the column indices of these columns. Now the column index set is $E(t) \cup E'(t)$, and for all $e \in E'(t)$, $\nu(T, e) = m$. Since $E(t) \cap E'(t) = \emptyset$, the mutual information of X_T and Y_E increases by

$$\begin{aligned}
& \sum_{t \in T} \left(\sum_{e \in E'(t)} \frac{1}{m(n-3)} \log \frac{\frac{1}{m(n-3)}}{\frac{1}{m} \frac{\nu(T,e)}{m(n-3)}} \right) \\
= & \sum_{t \in T} \left(\sum_{e \in E'(t)} \frac{1}{m(n-3)} \log \frac{\frac{1}{m(n-3)}}{\frac{1}{m} \frac{m}{m(n-3)}} \right) \\
= & \sum_{t \in T} \left(\sum_{e \in E'(t)} \frac{1}{m(n-3)} \log 1 \right) \\
= & \ 0
\end{aligned}$$

This shows the all-1 columns have no effect in $I(Z_C; Y_E)$. Similarly, the value $I(Z_C; Y_E)$ is not changed.

7.4 Experiments

7.4.1 Clustering algorithms

K-means clustering The input to the K-means algorithm is a set of points from some vector space. The number of clusters is specified beforehand. At the beginning of the algorithm, k random points are chosen as the initial means. At each iteration, the input is divided into k clusters by assigning each point to the closest mean. Then k new means are formed by computing the average of each cluster. The algorithm stops when the value of the objective function

$$\sum_{i=1}^{k} \sum_{t \in C_i} d(t, m_i)^2$$

does not change (m_i is the mean of cluster C_i, d is the Euclidean distance). It can be proven that the value of the objective function never increases during the algorithm.

We implemented two variants of the K-means algorithm. First we use binary vectors to represent trees. Let x_t be the vector corresponding to tree t. Every entry $(x_t)_i$ in x_t corresponds to a bipartition i induced by some internal edge in at least one tree in T; $(x_t)_i = 1$ if $i \in E(t)$; $(x_t)_i = 0$ otherwise. The mean of a cluster is the average of the binary vectors of the trees in the cluster; it does not necessarily represent a tree.

In the other variant, we use the strict consensus of each cluster as its mean. Trees in T are assigned to k clusters at random at the beginning; the k means are then calculated from these clusters. At each iteration, clusters are formed by placing trees with whatever mean they are closest to by RF distance. To get the means of the clusters in each iteration, we take the strict consensus tree. The algorithm quits when the objective function, in this case the sum of the distances from the trees to their closest mean, does not change.

Agglomerative clustering Agglomerative clustering starts by making each point in the input its own cluster. Iteratively, the two most similar clusters are chosen according to some similarity criterion, and are merged into a new cluster to replace the original two. The algorithm quits when some criterion is reached, and outputs the remaining clusters. In our experiment we use the number of clusters as the stop criterion.

We didn't change the agglomerative clustering algorithm. The pairwise distance used is again RF distance. The similarity measures are as follows:

1. Minimum pairwise distance: merge two clusters C_1 and C_2 that minimize $\min_{t_1 \in C_1, t_2 \in C_2} d(t_1, t_2)$.

2. Maximum pairwise distance: merge two clusters C_1 and C_2 that minimize $\max_{t_1 \in C_1, t_2 \in C_2} d(t_1, t_2)$.

3. Average pairwise distance: merge two clusters C_1 and C_2 that minimize $\frac{1}{|C_1||C_2|} \sum_{t_1 \in C_1, t_2 \in C_2} d(t_1, t_2)$.

When using the first and second similarity measures, the algorithms are called *single linkage* and *complete linkage*, respectively.

Settings for the experiment We use $2, 3, \ldots, 10$ clusters for both Agglomerative clustering (`Agg`) and K-means clustering (`Kmeans`). We also use the strict consensus trees of the clusters produced by the complete linkage for agglomerative clustering as the starting means in the K-means algorithm (`KmAgg`). The motive is to avoid being trapped in some local optimum due to the random effect in choosing starting means. See Table 7.1 for details.

7.4.2 Datasets

We obtained three datasets: `Camp` [20, 50], `Caesal` [87], and `PEVCCA` for our empirical study.

- The `Camp` dataset is obtained using the `GRAPPA` [52] software to reconstruct the breakpoint phylogeny of the *Campanulaceae* family (see [52] for an explanation of the breakpoint phylogeny). The dataset contains 216 trees on 13 leaves. The strict consensus tree for this dataset is 60% resolved.

- The `Caesal` dataset is obtained by maximum parsimony searches of the trnL-trnF intron and spacer regions of chloroplast genome from the *Caesalpinia* family. The dataset has 450 trees on 51 leaves. The strict consensus tree for this dataset is 77% resolved.

- The `PEVCCA` dataset is obtained by maximum parsimony searches of the small subunit ribosomal RNA sequences [82]; the dataset consists of 5630 trees on 129 leaves divided into 78 phylogenetic islands. `PEVCCA` stands for *Porifera* (sea sponges), *Echinodermata* (sea urchins, sea cucumbers), *Vertebrata* (fish, mammals, reptiles), *Cnidaria* (jellyfish), *Crustacea* (crabs, lobsters, shrimp), and *Annelida* (roundworms). The `PEVCCA1` dataset contains 168 most parsimonious trees of `PEVCCA` (1 island). The strict consensus tree for this dataset is 77% resolved. The `PEVCCA2` dataset includes the next best trees as well, for a total of 654 trees (5 islands). The strict consensus tree is 72% resolved.

Table 7.1: Clustering algorithms used in the experiments.

`Kmeans`	K-means using the strict consensus of a cluster as the mean.
`Agg 0`	Single-linkage agglomerative clustering.
`Agg 1`	Complete-linkage agglomerative clustering.
`Agg 2`	Agglomerative clustering using average distance.
`KmAgg`	`Kmeans` using `Agg 1` as starting means (not shown in the figures).
`KmVec`	K-means using the bipartition vector.
`PhyIsl`	Phylogenetic islands.
`1Clu`	One-cluster clustering: putting all trees in the same cluster.

7.4.3 Comparison of different algorithms

We compute the parameters in Table 7.2 for each clustering $\mathbf{C}$ produced by the algorithms being tested. The results are in Figures 7.1, 7.2, 7.3, 7.4, 7.5, 7.6, 7.7, and 7.8. See Table 7.2 for the legend (`KmAgg` is not shown in the figures since it has very a similar outcome to `Agg 1`). We add minus signs in front of those parameters when we favor larger values; therefore a lower value in the y-direction is always more favorable.

Comparison of methods

1. The `Caesal` dataset: Most of the time the `Kmeans` clustering has the worst performance of all the methods. With a large enough number of clusters (5 or above), the `KmVec` algorithm can have very good scores in parameters other than `L1`, `L2`, `Linf` and `KL`, but has suboptimal scores in these information-loss measures. The `Agg 0` algorithm (single link-

Table 7.2: Clustering parameters for the experiments.

`Linf`	the L_∞ distance.
`L1`	the L_1 distance.
`L2`	the L_2 distance.
`KL`	the Kullback-Leibler distance.
`DMI`	the decrease in mutual information (information bottleneck).
`maxdiam`	$W(\mathbf{C};\lambda)$, maximum clustal diameter.
`wtddiam`	$W(\mathbf{C};\mathrm{diam})$, weighted sum of diameter.
`minspec`	$W(\mathbf{C};\lambda)$, minimum clustal specificity.
`wtdspec`	$W(\mathbf{C};\mathrm{diam})$, weighted sum of specificity.
`logminden`	$W(\mathbf{C};\lambda)$, base-10 logarithm of minimum clustal density.
`logwtdden`	base-10 logarithm of weighted sum of density.

age) has very unsatisfying scores for all parameters, and increasing the number of clusters provides little improvement. Among the other two agglomerative clustering methods, `Agg 1` (complete linkage) has better overall performance for all parameters than `Agg 2`, which is better than `KmVec` in all information loss measures except with 10 clusters, `Linf` distance. The `PhyIsl` clustering is the same as the optimal two-cluster clustering.

It is interesting to note by increasing the number of clusters, `Kmeans` and `KmVec` can become worse. Since increasing the number of clusters in agglomerative clustering means refinement of the clustering by dividing some clusters, agglomerative clustering has better score in the monotone parameters; however the K-means clustering generally does not have the refinement relationship as we increase the number of clusters.

2. The `PEVCCA1` dataset: In this dataset and the next dataset (`PEVCCA2`), `L1`, `L2`, and `Linf` distances are uninformative: they always return values

close to or equal to the maximally allowed value. This is due to the relatively low density of each cluster, causing many trees to be in $B(\mathbf{C})$ but not in T, and thus contribute to the distance. However `KL` is very informative. There is only one phylogenetic island.

In this dataset, all clustering methods have similar performance for all parameters. When the number of clusters increase, all the parameters improve.

3. The `PEVCCA2` dataset: `Kmeans` is inferior to the performance of `Agg 1` and `Agg 2`, but better than `Agg 0`. `Agg 1`, and `Agg 2` have similar performance. When the number of clusters is low, `Agg 2` has better scores than `Agg 1`; when the number of clusters is high (5 or more), `Agg 1` and `Agg 2` have similar performance. `KmVec` can be as good as `Agg 1` and `Agg 2` until the number of clusters is 7 or more, where its performance becomes suboptimal.

   The performance of `PhyIsl` is very bad for all parameters considered, when compared to all other methods.

4. The `Camp` dataset: We applied `Agg 1` to this dataset. The dataset contains 216 trees out of 315 refinements of the strict consensus, which means the density is high. When we try to cluster the dataset, the specificity of the consensus trees improves slightly, but the density drops dramatically. This suggests that one cluster is sufficient for this dataset (the input trees are scattered uniformly in the cluster); that agglomerative clustering, by illustrating this fact, is robust. See Figure 7.9 for further evidence.

To summarize, `Agg 1` and `Agg 2` have the best overall performance. Both `Kmeans` and `KmVec` are unreliable, and `Agg 0` and `PhyIsl` tend to have worse performance.

Correlation of the parameters We make the following observations:

1. When comparing the values of different parameters, we find that all parameters are more or less correlated: in the `Caesal`, `PEVCCA1`, and `PEVCCA2` datasets, usually improving in one parameter means improving in all the other parameters. Major exceptions are `DMI` vs. other parameters in the `Camp` dataset, and `logminden` vs. other parameters in the `Caesal` dataset.

2. Parameters `L1`, `L2`, and `Linf` are not informative in `PEVCCA1` and `PEVCCA2` datasets: while significant changes are made in other parameters when we vary the clustering methods and number of clusters, these parameters always have (close to) the maximum value. This is due to the nature of sparse clusters in these two datasets. Since we tend to have very sparse clusters with the given numbers of clusters in these datasets, we can expect almost all trees with nonzero probability in the clustering distribution are not in the input tree. As a result, these distances are very high – numerically the difference between the actual distance and the maximum value is negligible.

3. For every biological criterion – the specificity, the diameter, and the density – the weighted-sum version and the extremum version can be different in the `Caesal`, `PEVCCA1`, and `PEVCCA2` datasets: usually the weighted-sum version has a smoother line in the figure. Furthermore,

in the Caesal and PEVCCA1 dataset, KL are more correlated with the extremum version of the biological criteria, while DMI is more correlated with the weighted-sum version of biological criteria. The difference between KL and DMI are smaller in the PEVCCA2 dataset.

4. The comparison in Caesal, PEVCCA1, and PEVCCA2 here does not determine which of KL and DMI is a better criterion. However, the difference between KL and DMI are greater in the Camp dataset. As the discussion of Camp dataset shows, any clustering with a small increase in the number of clusters should have very small improvement when compared to 1-clustering. This is not the case for DMI, which shows significant improvement as the number of clusters increase. Intuitively, clustering algorithms group trees with similar sets of bipartitions (edges) together; if we regard trees as "documents" and bipartitions as "words" as DMI suggests, clustering will always improve the information loss.

   We can draw two conclusions: (1) DMI is not designed for phylogenetic tree clustering, and can fail for certain datasets, (2) but for most real datasets, DMI is as good a criterion as KL.

7.4.4 Comparing clustering outputs to single-tree consensus

In this section we compare the outputs of clustering to the single-consensus approach. The comparison is done using Caesal, PEVVCA1, and PEVCCA2. In each dataset, we compare the output of Agg 1 with the strict consensus trees of the whole dataset. The number of clusters is determined by finding the number where the improvement starts to diminish; we use 3 clusters for Caesal and PEVCCA1, and 5 clusters for PEVCCA2. The results are in Table 7.3.

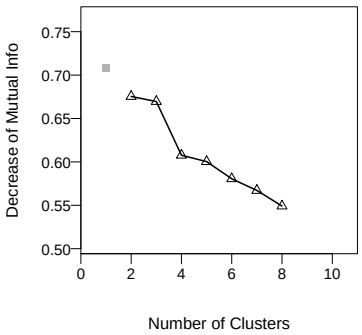
In each of the datasets, the strict consensus trees of each cluster is much more resolved than the strict consensus of the whole dataset. The `Caesal` dataset has one large cluster (cluster 2), one medium cluster (cluster 1), and one small cluster (cluster 3). The small cluster is sparse: it has more refinements than the medium cluster and has relatively few numbers of trees, suggesting it is a collection of outliers in the whole set of trees. Similarly, cluster 2 in the `PEVCCA1` dataset and cluster 3 and 5 in the `PEVCCA2` dataset are sparse clusters.

We remove these sparse clusters from the dataset. The percentage of trees dropped from `Caesal`, `PEVCCA1`, and `PEVCCA2` are 4%, 21.4%, and 14.4%, respectively. The specificity of the strict consensus of `Caesal`, `PEVCCA1`, and `PEVCCA2` have increased to 85.4%, 81.7%, and 75.4%, respectively. The result suggests the `Caesal` dataset is dominated by two major clusters (cluster 1 and 2) that are closer to each other than cluster 3; the small amount of increase of the specificity in `PEVCCA1` and `PEVCCA2` suggests the larger clusters are remote to each other.

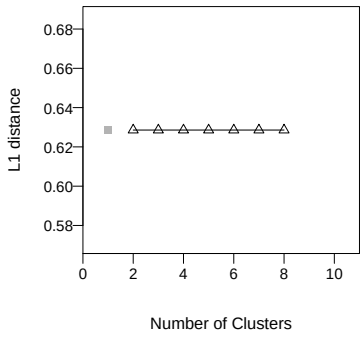
Table 7.3: Comparison of the clustering approach and the single-consensus approach. We use `Agg 1` with 3 clusters for `Caesal` and `PEVCCA1`, and 5 for `PEVCCA2`. In the number of edges field, the parenthesized value is the specificity. The "numtrees" and the "numref" field are the number of trees in the cluster and the refinements of the strict consensus of the cluster, respectively. The "1clu" row in each dataset corresponds to the strict consensus of the whole set of trees.

`Caesal`			
KL(`Agg 1`, 5 clusters)=1.449269			
KL(1 cluster)=9.790346			
clu#	numtrees	specificity	numref
1	108	89.6%	243
2	324	87.5%	729
3	18	89.6%	945
1clu	450	77.1%	8.037×10^6
`PEVCCA1`			
KL(`Agg 1`, 5 clusters)=23.553030			
KL(1 cluster)=45.456491			
clu#	numtrees	specificity	numref
1	94	92.1%	5.473×10^7
2	36	89.7%	2.846×10^{12}
3	38	92.1%	1.148×10^6
1clu	168	77.0%	9.264×10^{21}
`PEVCCA2`			
KL(`Agg 1`, 5 clusters)=21.972959			
KL(1 cluster)=53.405270			
clu#	numtrees	specificity	numref
1	114	92.6%	1.148×10^7
2	235	88.1%	7.795×10^{11}
3	6	93.7%	99225
4	211	87.3%	1.465×10^{12}
5	88	86.5%	2.110×10^{10}
1clu	654	72.2%	1.021×10^{26}

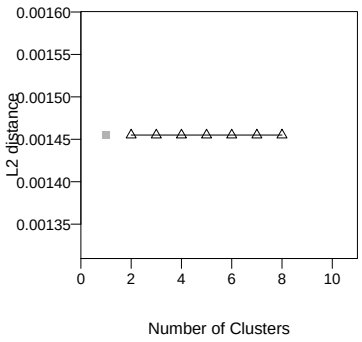
Number of clusters vs. `KL`



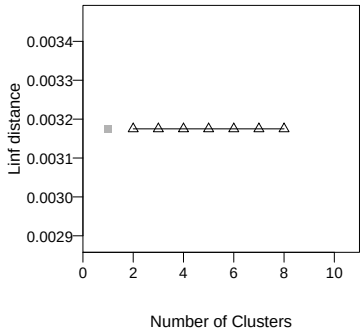
Number of clusters vs. `DMI`



Number of clusters vs. `L1`

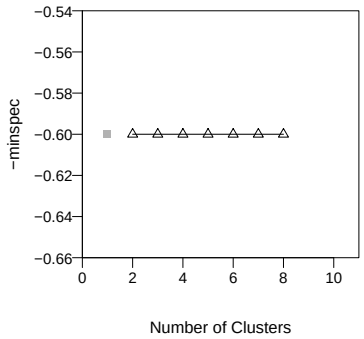


Number of clusters vs. `L2`

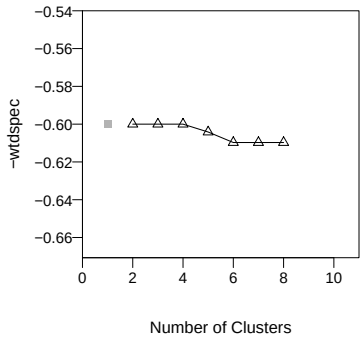


Number of clusters vs. `Linf`

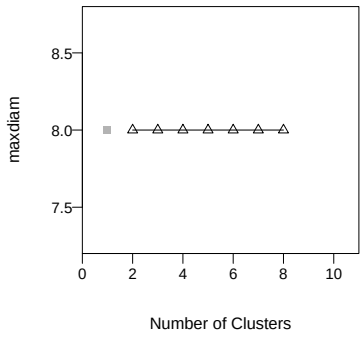
Figure 7.1: Results (statistical criteria) of the clustering experiment using the `Camp` dataset. See Section 7.4.1 for details.



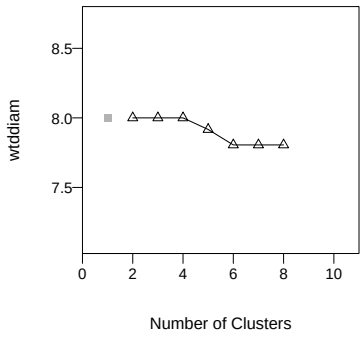
Number of clusters vs. `minspec`



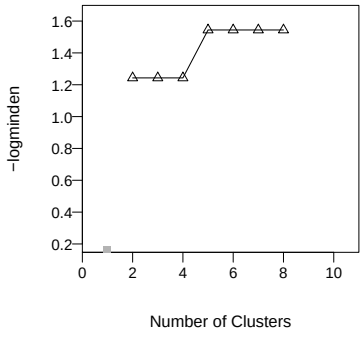
Number of clusters vs. `wtdspec`



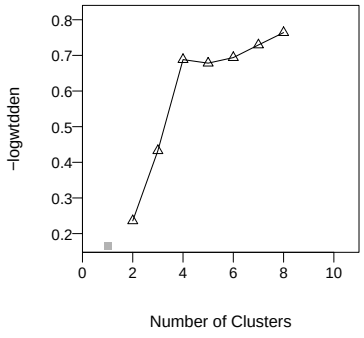
Number of clusters vs. `maxdiam`



Number of clusters vs. `wtddiam`

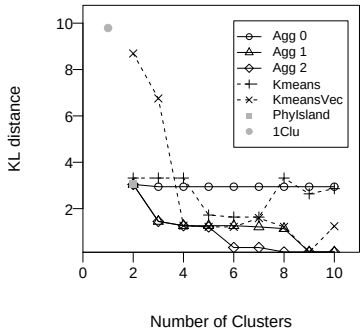


Number of clusters vs. `logminden`

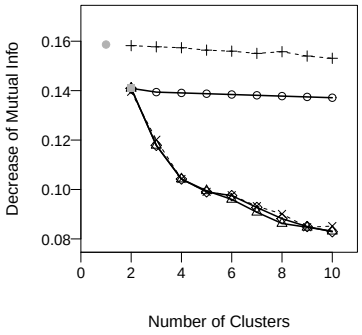


Number of clusters vs. `logwtdden`

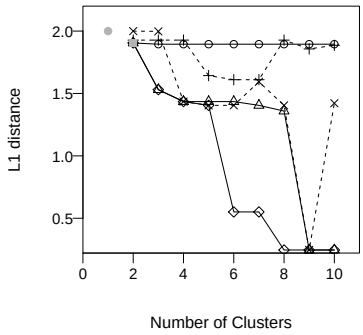
Figure 7.2: Results (biological criteria) of the clustering experiment using the `Camp` dataset. See Section 7.4.1 for details.



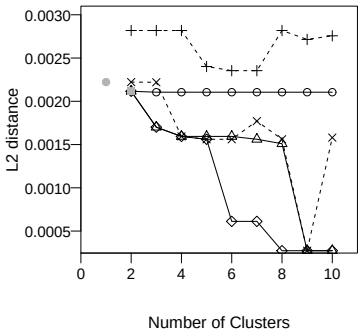
Number of clusters vs. `KL`



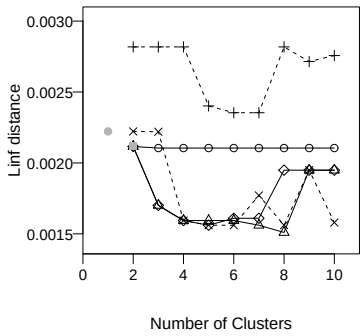
Number of clusters vs. `DMI`



Number of clusters vs. `L1`

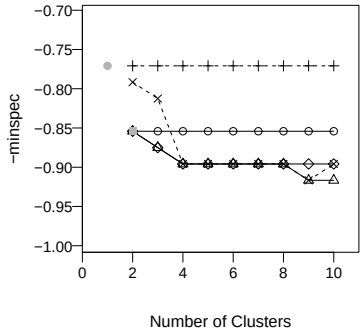


Number of clusters vs. `L2`

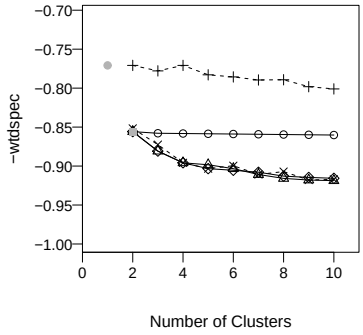


Number of clusters vs. `Linf`

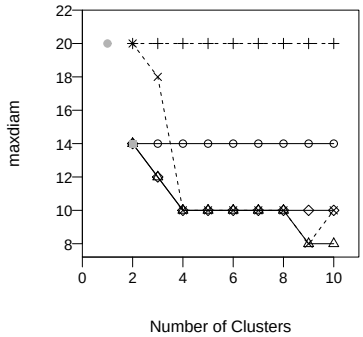
Figure 7.3: Results (statistical criteria) of the clustering experiment using the `Caesal` dataset. See Section 7.4.1 for details.



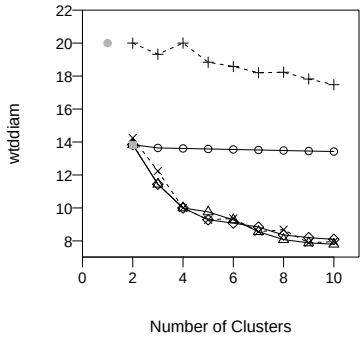
Number of clusters vs. `minspec`



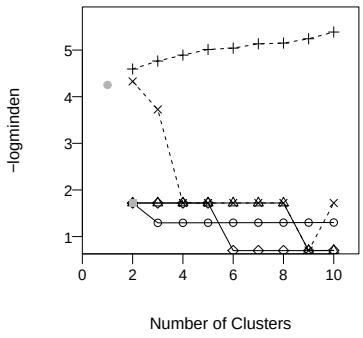
Number of clusters vs. `wtdspec`



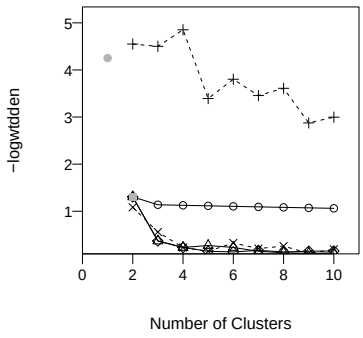
Number of clusters vs. `maxdiam`



Number of clusters vs. `wtddiam`

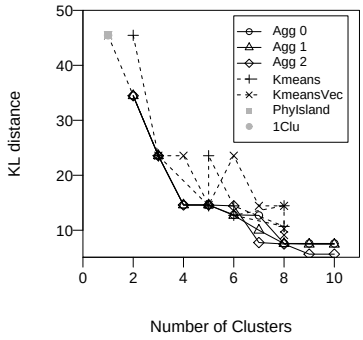


Number of clusters vs. `logminden`

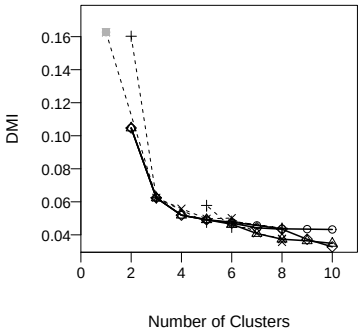


Number of clusters vs. `logwtdden`

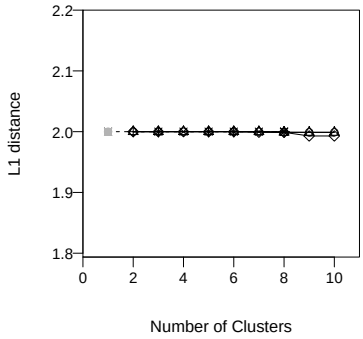
Figure 7.4: Results (biological criteria) of the clustering experiment using the `Caesal` dataset. See Section 7.4.1 for details.



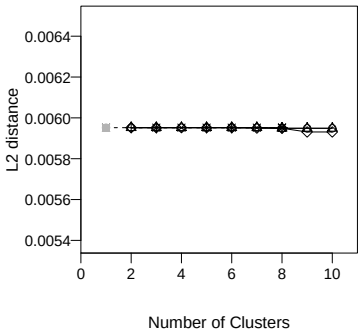
Number of clusters vs. `KL`



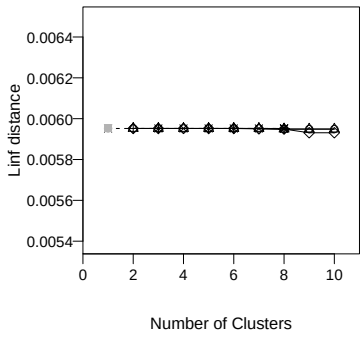
Number of clusters vs. `DMI`



Number of clusters vs. `L1`

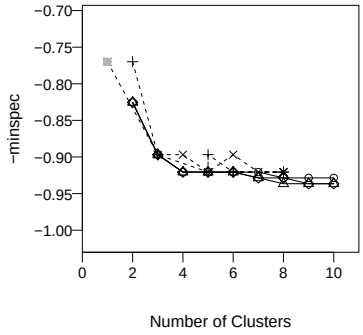


Number of clusters vs. `L2`

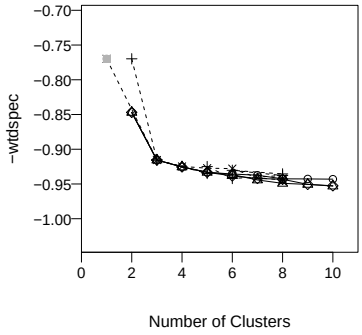


Number of clusters vs. `Linf`

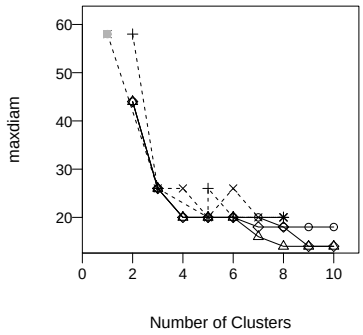
Figure 7.5: Results (statistical criteria) of the clustering experiment using the `PEVCCA1` dataset. See Section 7.4.1 for details.



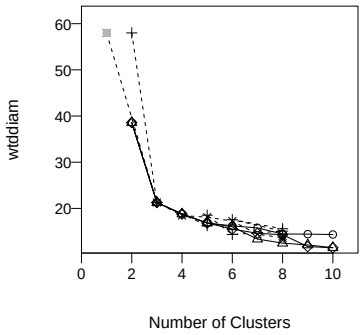
Number of clusters vs. `minspec`



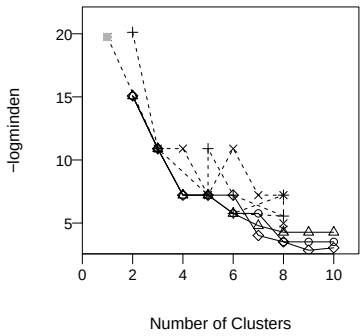
Number of clusters vs. `wtdspec`



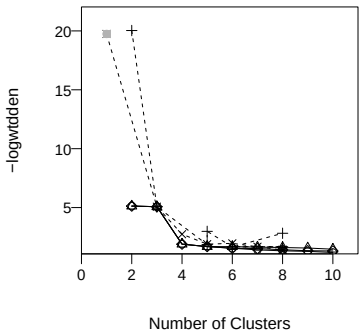
Number of clusters vs. `maxdiam`



Number of clusters vs. `wtddiam`

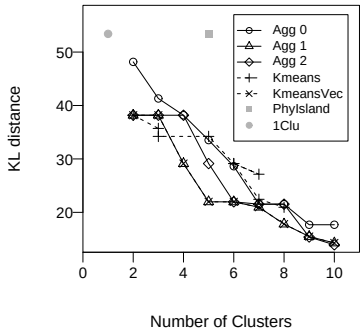


Number of clusters vs. `logminden`

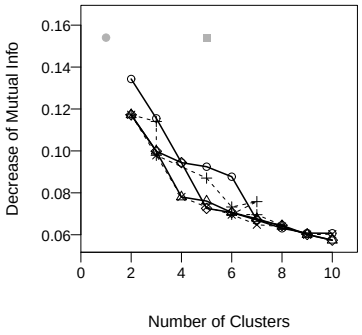


Number of clusters vs. `logwtdden`

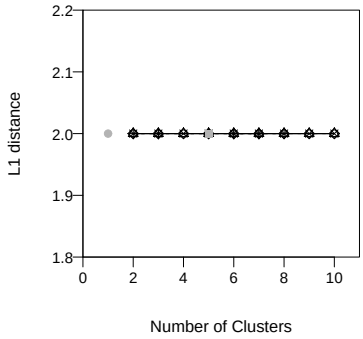
Figure 7.6: Results (biological criteria) of the clustering experiment using the `Caesal` dataset. See Section 7.4.1 for details.



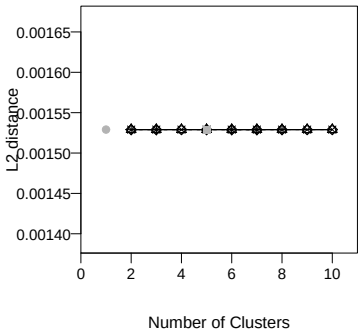
Number of clusters vs. `KL`



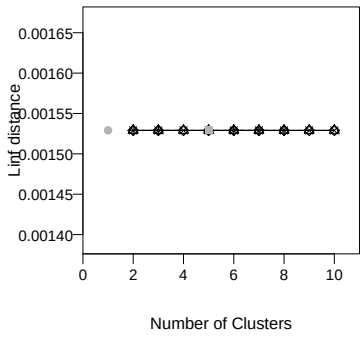
Number of clusters vs. `DMI`



Number of clusters vs. `L1`

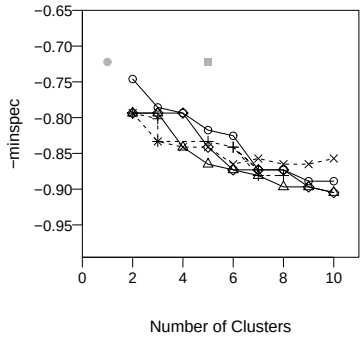


Number of clusters vs. `L2`

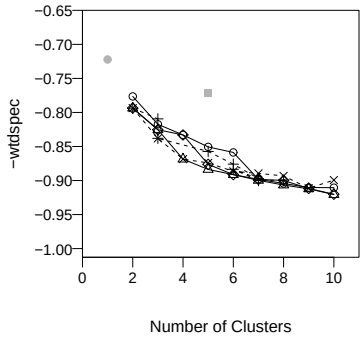


Number of clusters vs. `Linf`

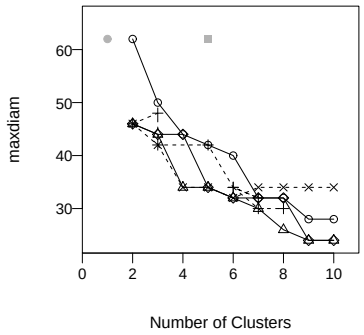
Figure 7.7: Results (statistical criteria) of the clustering experiment using the `PEVCCA2` dataset. See Section 7.4.1 for details.



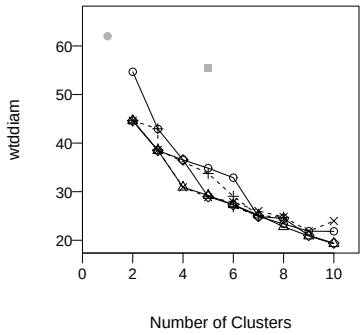
Number of clusters vs. `minspec`



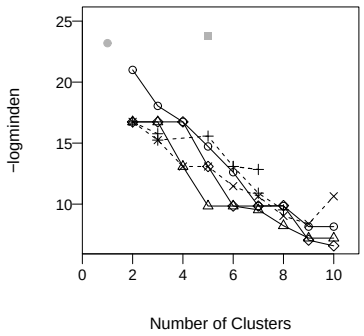
Number of clusters vs. `wtdspec`



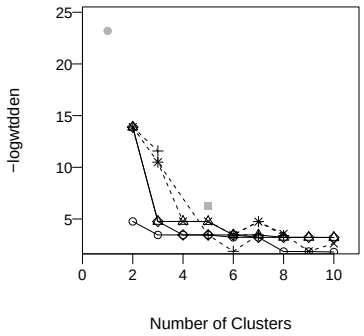
Number of clusters vs. `maxdiam`



Number of clusters vs. `wtddiam`



Number of clusters vs. `logminden`



Number of clusters vs. `logwtdden`

Figure 7.8: Results (biological criteria) of the clustering experiment using the `PEVCCA2` dataset. See Section 7.4.1 for details.

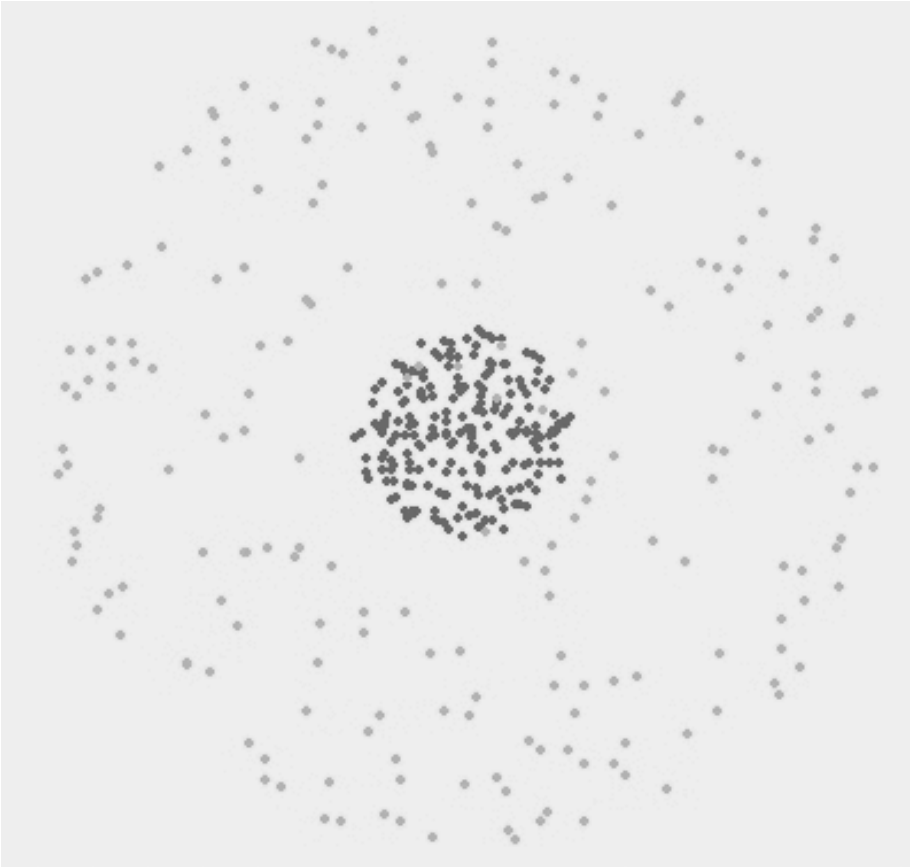


Figure 7.9: Three-dimensional embedding of the Campanulaceae dataset (Courtesy Nina Amenta and Jeff Klingner at the University of Texas at Austin). The figure is generated using the XGvis software [14]. The dark dots correspond to the input trees. We also add random trees to the dataset to give a sense about how the input trees are scattered with respect to the whole tree space; random trees are shown as light dots. Inspecting the output of XGvis shows the input trees are uniformly scattered on a smaller sphere, and the random trees are uniformly scattered on a larger, concentric sphere. Experience in using XGvis suggests the input trees are random with respect to one another in the bounding ball of the strict consensus (i.e. the smaller sphere).

Chapter 8

Conclusion

In this thesis I have studied the problem of large-scale phylogenetic analysis. I have proposed two approaches: using gene order (genome rearrangement) data as a new source of phylogenetic signal, and using clustering algorithms in the postprocessing stage.

In Chapter 3, I studied the distance-based approach for genome rearrangement phylogeny. I have proposed a generalization of the Nadeau-Taylor model that allows an arbitrary mix of different types of rearrangement events such as inversions and transpositions. Based on this model, several true evolutionary distances are proposed, including `Approx-IEBP` and `Exact-IEBP`, based on the breakpoint distance, and `EDE`, based on the inversion distance. When used with neighbor joining, the most popular distance-based tree reconstruction method, the accuracy of the inferred tree is greatly improved when compared with the old approach of using neighbor joining with either inversion or breakpoint distances. Among all these approaches, `NJ(EDE)`, neighbor joining with the `EDE` distance, has the best accuracy.

In Chapter 4, the variance of the genomic distances are studied. The main results include an analytical derivation approximating the variances of the breakpoint and `IEBP` (`Exact-IEBP`, `Approx-IEBP`) distances, and a numerical approach based on simulation data yielding formulas for the variances of the inversion and `EDE` distances. By modifying Weighbor, a variant of neigh-

bor joining designed for DNA sequence data, with the variances of `IEBP` and `EDE` distances, the accuracy of the trees reconstructed are even better than `NJ(EDE)`, especially when the dataset is close to saturation.

In Chapter 6, I examined the accuracy and running time of the fast parsimony-based tree reconstruction heuristics, including `MPBE-1` and `MPME`, for genome rearrangement data. Although having a higher running time (and `MPME` being limited by the 32-state limit in PAUP* 4.0), `MPME` and `MPBE-1` return trees with very low false negative rates. Furthermore, they are less affected by high evolutionary rates than distance-based methods. Among the three methods examined, `MPME` has the best overall accuracy.

Finally, in Chapter 7, I studied the problem of clustering in the space of phylogenetic trees. I have devised a framework that includes biological criteria such as the specificity (degree of resolution of the clusters) and the density of the cluster, and the statistical criterion called complexity *vs.* information loss. Using real biological datasets, the experimental study in this thesis suggests that when we use appropriate clustering algorithms, we lose less information than the traditional postprocessing methods including the single-tree consensus and the phylogenetic island method, and can identify outlier trees and improve the degree of resolution of the strict consensus. It also shows the information loss criterion is very informative regarding other biological criteria.

Bibliography

[1] E. Adams. N-trees as nestings: complexity, similarity and consensus. *J. Classif.*, 3:299–317, 1986.

[2] B. Allen and M. Steel. Subtree transfer operations and their induced metrics on evolutionary trees. *Annals of Combinatorics*, 5:1–13, 2001.

[3] K. Arrow. *Social choice and individual values.* Wiley Pub., 1963.

[4] K. Atteson. The performance of the neighbor-joining methods of phylogenetic reconstruction. *Algorithmica*, 25(2/3):251–278, 1999.

[5] D.A. Bader, B.M.E. Moret, and M. Yan. A fast linear-time algorithm for inversion distance with an experimental comparison. In *Lecture Notes in Computer Science No. 2125: Proc. 7th Workshop on Algs. and Data Structs. (WADS'01)*, pages 365–376, 2001.

[6] V. Bafna and P. Pevzner. Sorting permutations by transpositions. In *Proc. 6th Annual ACM-SIAM Symp. on Disc. Alg. (SODA'95)*, pages 614–623. ACM Press, 1995.

[7] M. Blanchette. Derange2. `http://www.cs.washington.edu/homes/blanchem/software.html`.

[8] M. Blanchette, G. Bourque, and D. Sankoff. Breakpoint phylogenies. In S. Miyano and T. Takagi, editors, *Genome Informatics*, pages 25–34. Univ. Acad. Press, 1997.

[9] M. Blanchette, M. Kunisawa, and D. Sankoff. Gene order breakpoint evidence in animal mitochondrial phylogeny. *J. Mol. Evol.*, 49:193–203, 1999.

[10] J. L. Boore, T. M. Collins, D. Stanton, L. L. Daehler, and W. M. Brown. Deducing arthropod phylogeny from mitochondrial DNA rearrangements. *Nature*, 376:163–165, 1995.

[11] W.J. Bruno, N.D. Socci, and A.L. Halpern. Weighted Neighbor Joining: a likelihood-based approach to distance-based phylogeny reconstruction. *Mol. Biol. Evol.*, 17:189–197, 2000.

[12] D. Bryant. A lower bound for the breakpoint phylogeny problem. In R. Giancarlo and D. Sankoff, editors, *Proc. 11th Ann. Symp. Combinatorial Pattern Matching (CPM'00)*, pages 235–247. Springer, 2000.

[13] D. Bryant, J. Tsang, P. Kearney, and M. Li. Computing the quartet difference between two trees. In *Proc. 11th ACM/SIAM Symp. on Disc. Alg. (SODA'00)*, pages 285–286, 2000.

[14] A. Buja, D.F. Swayne, M.L. Littman, and N. Dean. XGvis: interactive data visualization with multidimensional scaling, 1998. `http://www.research.att.com/areas/stat/xgobi/index.html\#xgvis`.

[15] P. Buneman. The recovery of trees from measures of dissimilarity. In F.R. Hobson, D.G. Kendal, and P. Tautus, editors, *Mathematics in the archaeological and historical sciences*. Edinburgh University Press, 1971.

[16] P. Buneman. The recovery of trees from measures of dissimilarity. In F. Hodson, D. Kendall, and P. Tautu, editors, *Mathematics in the Archae-*

ological and Historical Sciences, pages 387–395. Edinburgh University Press, 1971.

[17] A. Caprara. Formulations and hardness of multiple sorting by reversals. In *Proc. 3rd Int'l Conf. on Comput. Mol. Bio. (RECOMB'99)*, pages 84–93. ACM Press, NY, 1999.

[18] A. Caprara and G. Lancia. Experimental and statistical analysis of sorting by reversals. In D. Sankoff and J.H. Nadeau, editors, *Comparative Genomics*, pages 171–184. Kluwer Academic Publishers, 2000.

[19] W.J. Conover. *Practical Nonparametric Statistics, 3rd ed.* John Wiley & Sons, 1999.

[20] M.E. Cosner, R.K. Jansen, B.M.E. Moret, L.A. Raubeson, L.-S. Wang, T. Warnow, and S. Wyman. A new fast heuristic for computing the breakpoint phylogeny and a phylogenetic analysis of a group of highly rearranged chloroplast genomes. In *Proc. 8th Intl. Conf. on Intel. Sys. for Mol. Bio. (ISMB'00)*, pages 104–115. AAAI Press, 2000.

[21] B. DasGupta, T. Jiang, S. Kannan, M. Li, and E. Sweedyk. On the complexity and approximation of syntenic distance. In *Proc. 1st Int'l Conf. on Comput. Mol. Bio. (RECOMB'97)*, pages 99–108. ACM Press, NY, 1997.

[22] S.R. Downie and J.D. Palmer. Use of chloroplast DNA rearrangements in reconstructing plant phylogeny. In P. Soltis, D. Soltis, and J.J. Doyle, editors, *Molecular Systematics of Plants*, volume 49, pages 14–35. Chapman & Hall, 1992.

[23] V. Ferretti, J.H. Nadeau, and D. Sankoff. Original synteny. In *Proc. 7th Ann. Symp. Combinatorial Pattern Matching (CPM'96)*, pages 159–167. Springer, 1996.

[24] W. M. Fitch. Toward defining the course of evolution: minimal change for a specific tree topology. *Systematic Zoology*, 20:406–416, 1971.

[25] O. Gascuel. BIONJ: an improved version of the NJ algorithm based on a smple model of sequence data. *Mol. Biol. Evol.*, 14:685–695, 1997.

[26] O. Gascuel. Personal communication, April 2001.

[27] R. L. Graham, D. E. Knuth, and O. Patashnik. *Concrete Mathematics*. Addison-Wesley, 2 edition, 1994.

[28] S. Hannenhalli and P. Pevzner. Transforming cabbage into turnip (polynomial algorithm for genomic distance problems). In *Proc. 27th Annual ACM Symp. on Theory of Comp. (STOC'95)*, pages 178–189. ACM Press, NY, 1995.

[29] J. Hartigan. Minimum mutation fits to a given tree. *Biometrics*, 29:53–65, 1972.

[30] D. Hillis and J.J. Bull. An empirical test of bootstrapping as a method for assessing confidence in phylogenetic analysis. *Syst. Biol.*, 42:182–192, 1993.

[31] D. Hillis, J.J. Bull, M.E. White, M.R. Badgett, and I.J. Molineux. Experimental approaches to phylogenetic analysis. *Syst. Biol.*, 42:90–92, 1993.

[32] D. Hillis, J.J. Bull, M.E. White, M.R. Badgett M.R., and I.J. Molineux. Experimental phylogenies: generation of a known phylogeny. *Science*, 255:589–592, 1992.

[33] D. Huson. The tree library, 1999.

[34] D. Huson, S. Nettles, K. Rice, T. Warnow, and S. Yooseph. The hybrid tree reconstruction method. *J. Experimental Algorithmics*, 4:178–189, 1999. `http://www.jea.acm.org/`.

[35] R.K. Jansen. Personal communication, October 3 2000.

[36] T.H. Jukes and C.R. Cantor. Evolution of protein molecules. In H.N. Munro, editor, *Mammalian Protein Metabolism*, pages 21–132. Academic Press, New York, 1969.

[37] S. Kannan, T. Warnow, and S. Yooseph. Computing the local consensus of trees. In *Proc. 6th ACM/SIAM Symp. on Disc. Alg. (SODA'95)*, pages 68–77, 1995.

[38] H. Kaplan, R. Shamir, and R.E. Tarjan. Faster and simpler algorithm for sorting signed permutations by reversals. In *Proc. 8th Annual Symp. on Discrete Alg. (SODA'97)*, pages 344–351. ACM Press, NY, 1997.

[39] M. Kimura. A simple model for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences. *J. Mol. Evol.*, 16:111–120, 1980.

[40] J. Kleinberg and D. Liben-Nowell. The syntenic diameter of the space of N-chromosome genomes. In D. Sankoff and J.H. Nadeau, editors, *Comparative Genomics*, pages 185–197. Kluwer Academic Pubs., 2000.

[41] J. Klingner. Visualization of phylogenetic data. Department of Computer Science, University of Texas at Austin.

[42] S. Kullback. The Kullback-Leibler distance. *Am. Stat.*, 41:340, 1987.

[43] S. Kumar. Minimum evolution trees. *Mol. Biol. Evol.*, 15:584–593, 1996.

[44] D.R. Maddison. The discovery and importance of multiple islands of most parsimonious trees. *Systematic Zoology*, 40:315–328, 1991.

[45] T. Margush and F. McMorris. Consensus n-trees. *Bulletin of Mathematical Biology*, 43:239–244, 1981.

[46] S. Mathews and M. J. Donoghue. The root of angiosperm phylogeny inferred from duplicate phytochrome genes. *Science*, 286:947–950, 1999.

[47] MathWorks. *Matlab 6.1*, (2000). Natick, MA USA.

[48] F.R. McMorris. Axioms for consensus functions on undirected phylogenetic trees. *Math. Biosci.*, 74:17–21, 1985.

[49] F.R. McMorris and M.A. Steel. The complexity of the median procedurefor binary trees. In *Proceedings of the International Federation of Classification Societies*. Springer-Verlag, 1993.

[50] B.M.E. Moret, L.-S. Wang, T. Warnow, and S. Wyman. New approaches for reconstructing phylogenies based on gene order. In *Proc. 9th Intl. Conf. on Intel. Sys. for Mol. Bio. (ISMB'01)*, pages 165–173, 2001.

[51] B.M.E. Moret, S.K. Wyman, D.A. Bader, T. Warnow, and M. Yan. A new implementation and detailed study of breakpoint analysis. In *Proc.*

6th Pacific Symp. Biocomputing (PSB'01), pages 583–594. World Scientific Pub., 2001.

[52] B.M.E Moret, S.K. Wyman, D.A. Bader, T. Warnow, and M. Yan. A new implementation and detailed study of breakpoint analysis. In *Proc. 6th Pacific Symp. Biocomputing (PSB'01)*, pages 583–594, 2001.

[53] J.H. Nadeau and B.A. Taylor. Lengths of chromosome segments conserved since divergence of man and mouse. *Proc. Nat'l Acad. Sci. USA*, 81:814–818, 1984.

[54] L. Nakhleh, U. Roshan, K. St. John, J. Sun, and T. Warnow. Designing fast converging phylogenetic methods. In *Proc. 9th Intl. Conf. on Intel. Sys. for Mol. Bio. (ISMB'01)*, pages 190–198, 2001.

[55] National Institutes of Health National Center for Biotechnology Information. The NCBI microbial genome dataset. `http://www.ncbi.nlm.nih.gov/PMGifs/Genomes/new_micr.html`.

[56] G. Nelson. Cladistic analysis and synthesis: principles and definitions, with a historical not on Adanson's familles des plantes (1763-1764). *Syst. Zoology*, 28:1–21, 1979.

[57] G. W. Oehlert. A note on the delta method. *Amer. Statist.*, 46:27–29, 1992.

[58] R.G. Olmstead and J.D. Palmer. Chloroplast DNA systematics: a review of methods and data analysis. *Amer. J. Bot.*, 81:1205–1224, 1994.

[59] J.D. Palmer. Chloroplast and mitochondrial genome evolution in land plants. In R. Herrmann, editor, *Cell Organelles*, pages 99–133. Wein, 1992.

[60] I. Pe'er and R. Shamir. The median problems for breakpoints are NP-complete. *Elec. Colloq. on Comput. Complexity*, 71, 1998.

[61] C.A. Phillips and T. Warnow. The asymmetric median tree: a new model for building consensus trees. *Disc. App. Math.*, 71:311–335, 1996.

[62] L.A. Raubeson and R.K. Jansen. Chloroplast DNA evidence on the ancient evolutionary split in vascular land plants. *Science*, 255:1697–1699, 1992.

[63] D.F. Robinson and L.R. Foulds. Comparison of phylogenetic trees. *Math. Biosci.*, 53:131–147, 1981.

[64] A. Rokas and P. W. H. Holland. Rare genomic changes as a tool for phylogenetics. *Trends in Ecology and Evolution*, 15:454–459, 2000.

[65] N. Saitou and M. Nei. The neighbor-joining method: A new method for reconstructing phylogenetic trees. *Mol. Biol. & Evol.*, 4:406–425, 1987.

[66] D. Sankoff and M. Blanchette. Multiple genome rearrangement and breakpoint phylogeny. *J. Comp. Biol.*, 5:555–570, 1998.

[67] D. Sankoff and M. Blanchette. Probability models for genome rearrangements and linear invariants for phylogenetic inference. *Proc. 3rd Int'l Conf. on Comput. Mol. Bio. (RECOMB'99)*, pages 302–309, 1999.

[68] D. Sankoff and Nadia El-Mabrouk. Rearrangement and reconciliation. In D. Sankoff and J.H. Nadeau, editors, *Comparative Genomics*, pages 171–184. Kluwer Academic Publishers, 2000.

[69] D. Sankoff and J.H. Nadeau, editors. *Comparative Genomics : Empirical and Analytical Approaches to Gene Order Dynamics, Map Alignment and the Evolution of Gene Families*. Kluwer Academic Publishers, 2000.

[70] C. Seoighe et al. Prevalence of small inversions in yeast gene order evolution. *Proc. Natl. Acad. Sci. USA*, 97:14433–14437, 2000.

[71] M. Spencer, B. Bordalejo, L.-S. Wang, A.C. Barbrook, L.R. Mooney, P. Robinson, T. Warnow, and C.J. Howe. Analyzing the order of items in manuscripts of *The Canterbury Tales*, February 2003.

[72] K. St. John, L.-S. Wang, and T. Warnow. Computing phylogenetic islands. manuscript.

[73] K. St. John, T. Warnow, B.M.E. Moret, and L. Vawter. Performance study of phylogenetic methods: (unweighted) quartet methods and neighbor-joining. In *Proc. 12th Ann. Symp. Discrete Algs. (SODA'01)*, pages 196–205. SIAM Press, 2001.

[74] M. Steel, S. Becker, and A.W.M. Dress. Some simple but fundamental limits for supertree and consensus tree methods. *Sys. Biol.*, 42(2):363–368, (2000).

[75] M. Steel and D. Penny. Parsimony, likelihood, and the role of models in molecular phylogenetics. *Mol. Biol. Evol.*, 17(6):839–850, 2000.

[76] M. Steel and L.A. Szekely. Inverting random functions (II): explicit bounds for discrete maximum likelihood estimation, with applications. *SIAM J. Discr. Math.*, 15(4):562–575, 2002.

[77] C. Stockham, L.-S. Wang, and T. Warnow. Statistically based postprocessing of phylogenetic analysis using clustering. *Bioinformatics: Proceedings of the Tenth International Conference on Intelligent Systems for Molecular Biology (ISMB02)*, 18(Supp.1)):285–293, 2002.

[78] D. Swofford. *PAUP* 4.0.* Sinauer Associates Inc, 2001.

[79] D. Swofford, G. Olson, P. Waddell, and D. Hillis. Phylogenetic inference. In D. Hillis, C. Moritz, and B. Mable, editors, *Molecular Systematics*, chapter 11. Sinauer Associates Inc, 2 edition, 1996.

[80] N. Tishby, F. Pereira, and W. Bialek. The information bottleneck method. In *The 37th annual Allerton Conference on Communication, Control, and Computing*, 1999.

[81] De Montfort University. The Canterbury Tales Project. `http://www.cta.dmu.ac.uk/projects/ctp/index.html`.

[82] Y. Van de Peer, P. De Rijk, J. Wuyts, T. Winkelmans, and R. De Wachter. The european small subunit ribosomal RNA database. *Nucleic Acids Res.*, 28:175–176, (2000).

[83] L.-S. Wang. Improving the accuracy of evolutionary distances between genomes. In *Lec. Notes in Comp. Sci.: Proc. 1st Workshop for Alg. & Bio. Inform. (WABI'01)*, pages 175–188. Springer Verlag, 2001.

[84] L.-S. Wang. Genome rearrangement phylogeny using Weighbor. In *Lec. Notes in Comp. Sci.: Proc. 2nd Workshop for Alg. & Bio. Inform. (WABI'02)*, pages 112–125. Springer-Verlag, 2002.

[85] L.-S. Wang, R.K. Jansen, B.M.E. Moret, L.A. Raubeson, and T. Warnow. Fast phylogenetic methods for the analysis of genome rearrangement data: an empirical study. In *Proceedings of the Fifth Pacific Symposium of Biocomputing (PSB02)*, pages 524–535, 2002.

[86] L.-S. Wang and T. Warnow. Estimating true evolutionary distances between genomes. In *Proc. 33th Annual ACM Symp. on Theory of Comp. (STOC'01)*, pages 637–646. ACM Press, 2001.

[87] A. Weeks, L. Larkin, and B. Simpson. A chloroplast DNA molecular study of the phylogenetic relationships of members of the *Caesalpinia* group. In *Botany 2001 Abstracts*, page 164. Botanical Society of America, (2001).

[88] M. Wilkinson. Common cladistic information and its consensus representation: reduced adams and reduced cladistic consensus trees and profiles. *Sys. Biol.*, 43:343–368, 1994.

Vita

Li-San Wang was born in Taoyuan, Taiwan on April 28, 1972. He received his Bachelor of Science degree in June 1994, and his Master of Science degree in June 1996, both from the Department of Electrical Engineering, National Taiwan University. From July 1996 to May 1998 he served as a second lieutenant in the Republic of China (Taiwan) army. In September 1998 he entered the Graduate School of The University of Texas. He received his Master of Science degree in Computer Science in May 2000, and expects to receive his degree of Ph.D. in Computer Science in May 2003.

Mr. Wang's research interests include theory and practice of algorithms, computational phylogenetics, and other topics in computational biology and bioinformatics. He has published 11 peer-reviewed articles in internationally renowned conferences and journals since the year 2000. He is a member of ISCB and ACM.

Permanent address: 41 Hsin-Hsin Road
Ping-Chen City, Taoyuan Hsien
Taiwan 320

This dissertation was typeset with LaTeX[†] by the author.

[†]LaTeX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's TeX Program.